



Infor Interaction Advisor Administration Guide

Release 10.0.5.0

Important Notices

The material contained in this publication (including any supplementary information) constitutes and contains confidential and proprietary information of Infor.

By gaining access to the attached, you acknowledge and agree that the material (including any modification, translation or adaptation of the material) and all copyright, trade secrets and all other right, title and interest therein, are the sole property of Infor and that you shall not gain right, title or interest in the material (including any modification, translation or adaptation of the material) by virtue of your review thereof other than the non-exclusive right to use the material solely in connection with and the furtherance of your license and use of software made available to your company from Infor pursuant to a separate agreement, the terms of which separate agreement shall govern your use of this material and all supplemental related materials ("Purpose").

In addition, by accessing the enclosed material, you acknowledge and agree that you are required to maintain such material in strict confidence and that your use of such material is limited to the Purpose described above. Although Infor has taken due care to ensure that the material included in this publication is accurate and complete, Infor cannot warrant that the information contained in this publication is complete, does not contain typographical or other errors, or will meet your specific requirements. As such, Infor does not assume and hereby disclaims all liability, consequential or otherwise, for any loss or damage to any person or entity which is caused by or relates to errors or omissions in this publication (including any supplementary information), whether such errors or omissions result from negligence, accident or any other cause.

Without limitation, U.S. export control laws and other applicable export and import laws govern your use of this material and you will neither export or re-export, directly or indirectly, this material nor any related materials or supplemental information in violation of such laws, or use such materials for any purpose prohibited by such laws.

Trademark Acknowledgements

The word and design marks set forth herein are trademarks and/or registered trademarks of Infor and/or related affiliates and subsidiaries. All rights reserved. All other company, product, trade or service names referenced may be registered trademarks or trademarks of their respective owners.

Publication Information

Release: Infor Interaction Advisor 10.0.5.0

Publication Date: May 30, 2019

Document code: ia_10.0.5.0_iaag__en-us

Contents

About This Guide.....	8
Related Documents.....	8
Viewing Release Notes and Manuals Online.....	9
Contacting Customer Support.....	9
Printing This Document.....	10
Contacting Infor.....	10
Chapter 1: Introduction.....	11
Interaction Advisor.....	11
Chapter 2: Starting RT Administrator.....	12
Starting Real-Time Administrator.....	12
To Start Real-Time Administrator from the Start Menu.....	12
To Start Real-Time Administrator from your Web Browser.....	13
Chapter 3: Working with Real-Time Server Settings.....	14
Viewing Server Setting Details.....	14
Changing Server Setting Parameters.....	14
Chapter 4: Server Settings.....	17
Overview.....	17
Changing Server Settings.....	19
Saving Packages, Real-Time Miners, and Audit Trails in the RTDB.....	19
Restoring Lost Data.....	20
Reassigning Campaign Packages.....	20
Chapter 5: Deployment Groups.....	22
Deployment Group Support.....	22
Master-Slave Relationships.....	23
Failover Relationships.....	23
Accessing Deployment Group Settings.....	24

Deployment Group Settings.....	24
Master Real-Time Server Settings.....	25
Slave Real-Time Server Settings.....	25
Defining a Deployment Group.....	25
Redefining a Deployment Group.....	27
Adding a Slave to an Existing Deployment Group.....	27
Adding a Slave Containing Real-Time Miner Data.....	27
Deployment Group Shares Real-Time Miner Learnings.....	27
Deployment Group Does Not Share Real-Time Miner Learnings.....	28
Assigning a Slave the Role of Master.....	28
Splitting a Deployment Group.....	29
Removing a Slave.....	29
Designating Backup Servers.....	29
Chapter 6: HTTP.....	31
HTTP Settings.....	31
Chapter 7: Port Interface.....	32
Port Interface Settings.....	32
Changing Port Interface Settings.....	34
Chapter 8: Security.....	35
The Security Module.....	35
Security Certificates.....	35
Obtaining a server certificate.....	36
Changing Certificates.....	37
Chapter 9: Users and User Groups.....	38
Users.....	38
Privileges.....	38
User Settings.....	39
Defining Users.....	40
User Access Groups.....	42
User Access Group Settings.....	42
User Package Groups.....	42
User Package Group Settings.....	42
Procedures.....	43
Chapter 10: Data Access.....	45

Database Access Rights.....	45
Data Access Settings.....	46
OfferTracker.....	47
Deleting a Data Access Service.....	48
Adding a DB2 Data Access Service.....	48
Adding an Oracle Data Access Service.....	49
Modifying a Data Access Service.....	49
Chapter 11: BIO Access.....	50
BIO Service Settings.....	50
BIO Access Error Messages.....	52
Chapter 12: Java Adapter.....	54
Java Adapter Settings.....	54
Environment Variables.....	55
Chapter 13: Real-Time Miners.....	56
Accessing Real-Time Miner Settings.....	56
Real-Time Miner Settings.....	56
Creating Real-Time Miner Services.....	58
Modifying Real-Time Miner Services.....	58
Deleting Real-Time Miner Services.....	59
Storing Real Time Miners in an RTDB.....	59
Storing and Sharing Older Real-Time Miners.....	60
How Real-Time Miners Work in a Deployment Group.....	60
With Store RT Miner Models in Database Enabled.....	61
With Store RT Miner Models in Database Disabled.....	61
Chapter 14: Recommendation Models.....	62
Recommendation Model Settings.....	62
Chapter 15: Campaign Packages.....	64
Loading Packages at Deployment.....	64
Changing Minimum Package Load Settings.....	65
Campaign Packages Settings.....	65
User Editing Session.....	65
Copy to Server.....	66
Campaign Package Deployment Tools.....	66

Chapter 16: Package Archiving	70
Auto Package Archiving	70
Manual Package Archiving	71
Chapter 17: Logging	72
The CAMPAIGNPACKAGE ERROR Log	72
The SYSTEM_TRACE Log	73
Logging Settings	73
Creating a Log Record	74
Defining Custom Logs	74
Modifying Logs	75
Chapter 18: Performance Monitoring Tools	77
Performance Monitoring on Windows	77
Counters	77
Performance Monitoring on Linux	80
Chapter 19: Troubleshooting	82
Real-Time Server Setup	82
Campaign Package Deployment	82
Data Access	82
RT Client Application Setup	83
Client Connectivity	83
Database Permissions	83
Memory Usage	83
Windows Error Reporting	84
Chapter 20: Debugging	85
Debugging Settings	85
Changing Debugging Settings	86
Chapter 21: Admin Variables	87
To View Admin Variables	87
To Add Admin Variables	87
To Modify Admin Variables	88
To Delete Admin Variables	88
Glossary	89

Index.....100

About This Guide

This guide explains how to use Real-Time Administrator, a browser-based application for IT professionals, to configure communication between Infor Real-Time Server and other elements of the campaign environment.

Every Real-Time Server installation is accompanied by a Real-Time Administrator installation. When you use multiple Real-Time Servers on hosts connected to each other over a local-area or wide-area network, you can use any Real-Time Administrator to configure any Real-Time Server on the same network.

Related Documents

Documentation for Interaction Advisor includes the following manuals:

- *Installation Guide*, which explains how to install Real-Time Server, Real-Time Studio, Interaction Advisor Manager, Real-Time Database, and other related components.
- *Real-Time Implementation Guide*, a three part guide. Part I explains how to integrate Real-Time with a variety of customer touch points. Part II documents the Real-Time Database, details important performance aspects of Interaction Advisor, and includes helpful hints for developing and testing packages. Part III explains how to implement various features of Interaction Advisor according to a number of business/use case scenarios.

Documentation for Interaction Advisor also includes the following online help systems:

- *Real-Time Studio Online Help*
- *Interaction Advisor Manager Online Help*
- *Interaction Advisor JavaScript Reference* - combines the following kinds of information in a single online system:
 - A primer of relevant JavaScript basics and objects.
 - The Real-Time Campaign Objects JavaScript reference.
 - Real-Time Application Library online help, which explains how to use the provided templates and the Application Library, a set of objects and methods that ease the DB interactions within your applications.
 - Real-Time COM Adapter help, which explains how to use COM objects to access data that cannot be defined as campaign elements in the campaign package.
 - Real-Time Java Adapter help, which explains how Real-Time Server's Java Adapter uses Java classes from JavaScript functions you write using Interaction Advisor Manager.

To access this information, choose **JavaScript Reference** from the Help menu in Real-Time Studio.

The following documents are not provided with Interaction Advisor. They are for Marketing which can be used by itself or in conjunction with Interaction Advisor to manage campaigns and activities:

- *Infor Omni-channel Campaign Management Installation Guide*
- *Infor Omni-channel Campaign Management Data Mart Implementation Guide*
- *Infor Omni-channel Campaign Management Implementation Guide*
- *Infor E-Mail Marketing Administrator's Guide*

For information about integrating Infor Omni-channel Campaign Management with Infor Interaction Advisor, see Chapter 12 in the *Infor Omni-channel Campaign Management Implementation Guide*.

Viewing Release Notes and Manuals Online

You can view our product documentation and Release Notes on any computer that has Acrobat Reader running.

When you install documentation on Windows, the installation wizard adds shortcuts to the **Start** menu for easy viewing access. To view the installed documentation for Interaction Advisor, navigate to:

Start > Programs > Epiphany > Interaction Advisor > Documentation > Language

The documentation is installed in a directory that matches language. (The default location of this directory in Windows is `C:\Program Files\Epiphany\RT\Doc\<locale>`.)

On Linux platforms, the documentation installs to `<InstallationDirectory>/Doc/<locale>`. To view the documentation, first start Acrobat Reader, and then open the PDF file.

Installation instructions are included in the Installation Guide.

Contacting Customer Support

Before you call, please have the following information available:

- Infor Interaction Advisor version number.
- Infor Interaction Advisor platform type; your Windows/version, or your Linux type and version.
- Database type (Oracle, SQL Server, DB2 including version and patch level).
- Client platform (Windows/version).
- Client browser version.
- The location of the Infor Interaction Advisor database.
- The sequence of events leading to the problem plus any program or error messages displayed. If possible, have the program running when you call Customer Support.

To contact Customer Support:

You may contact the Infor Customer Support center by submitting your incident via the web 24x7 at <http://www.inforxtreme.com>, or by placing a call during our scheduled business hours. For a complete listing of our support centers with web addresses and phone numbers, access our support site at <http://www.inforxtreme.com>.

Printing This Document

If you print this document, the best quality is achieved by printing to a PostScript driver. Other drivers may not reproduce the screen shots as accurately.

This document is designed to be printed on two sides of the page. If your printer isn't configured for duplex printing, you may find a blank page at the end of some chapters. This is normal.

Note: Where printing reports, data, etc. from the application is concerned, IA does not support the IE Print functionality.

Contacting Infor

If you have questions about Infor products, go to Infor Concierge at <https://concierge.infor.com/> and create a support incident.

If we update this document after the product release, we will post the new version on the Infor Support Portal. To access documentation, select **Search > Browse Documentation**. We recommend that you check this portal periodically for updated documentation.

If you have comments about Infor documentation, contact documentation@infor.com.

Chapter 1: Introduction

Infor provides CRM software solutions that increase profitability at the largest consumer-oriented companies by making every customer interaction intelligent. With customers in financial services, insurance, retail, communications, and travel & leisure – including nearly 35 percent of the Fortune 100 – Infor powers deep customer insights and optimizes each relationship from both a revenue generation and customer retention viewpoint.

With a suite of blended marketing, sales and service solutions, Infor enables global organizations to align touchpoints, processes and technologies around the most valuable enterprise asset – the customer. Built on the industry's most advanced, service-oriented architecture, Infor solutions address problems that span business silos, departmental functions and geographic locations, and result in rapid, measurable ROI. With worldwide headquarters in New York, New York, Infor serves customers in more than 40 countries worldwide.

Interaction Advisor

The application uses customer data from across the enterprise and delivers inbound marketing offers to various touch points during the customer interaction. Controlled by a marketing professional through an intuitive web-based interface, the application provides a framework for creating, executing, modifying, and monitoring inbound marketing campaigns.

Chapter 2: Starting RT Administrator

This chapter describes how to start Real-Time Administrator from the Windows Start menu, from your Web browser.

Starting Real-Time Administrator

You can start Real-Time Administrator from the Windows **Start** menu or by specifying a URL in your browser.

Note: Start the Real-Time Server you want to work with before you start Real-Time Administrator.

To log on to Real-Time Administrator for the very first time, use the name and password of the Real-Time administrative user that was selected during Real-Time Server installation.

To Start Real-Time Administrator from the Start Menu

Note: If AD SSO has been enabled, RT Administrator must be started from your web browser according to the instructions in the next section.

- 1 In the Start menu, select **Programs > Epiphany > Real-Time > Real-Time Administrator**. Your default browser starts.
For Windows 2012, select **Start > Apps > Real-Time Administrator**.
- 2 Enter the name or IP address of the Real-Time Server you want to work with. If Infor S&S Single Sign-On (SSO) is enabled, you must enter the full DNS name of the server.
- 3 Enter your defined user name and password.
- 4 If the Secure Socket Layer (SSL) option is enabled for the target Real-Time Server and you want to communicate with Real-Time Administrator via a secure socket connection, select **Use Secure Port** and click **Advanced** to display the default port settings.
- 5 Change the secure port from its default setting (7203) to the number of the secure port you want to use.
- 6 If you do not want to use SSL communications, clear **Use Secure Port** and change the HTTP port from its default Setting (7201) to the number of the port you want to use.
- 7 Click **OK**.

To Start Real-Time Administrator from your Web Browser

- 1 Enter a URL that identifies RT Administrator and the Real-Time Server you want to work with in the form:

```
http://RT-Server-identifier:port/Admin/default.htm
```

RT-Server-identifier

The name or IP address of the target Real-Time Server. If S&S SSO is enabled, you must use the full DNS name of the server.

This Real-Time Server must be running when you start Real-Time Administrator.

port

The port that Real-Time Server monitors to receive Internet messages.

To use SSL communications, specify the secure port. Default is 7203. Otherwise, specify the Real-Time Server's HTTP port. Default is 7201.

If AD SSO has been enabled, the URL should be in the form:

```
http://RT-Server-identifier:port/Admin
```

Where:

RT-Server-identifier: The name or IP address of the target Real-Time Server.

port The port associated with the IIS authentication site.

- 2 In the resulting dialog box, enter a user name and password that the target Real-Time Server is configured to accept; the user must have Administrator rights.

To log in to Real-Time Administrator for the first time after Real-Time Server installation, enter the user name specified during installation. No password is required for the initial login since the authentication option is set to **none** by default.

For a production system, Infor strongly recommends enabling either operating system or Single Sign-On (SSO) authentication.

See [Server Settings](#) on page 17 and the appropriate guide available for Single Sign-On Server.

Note: Single Sign-On Server is available through Sales and Service - another Infor Enterprise Marketing Suite application, and also through Infor Ming.le. For the latest information about the version of Sales and Service or Infor Ming.le which is supported with Interaction Advisor, see the Platform Support Matrix document available on the Infor Customer Support web site: <http://www.inforxtreme.com>.

- 3 (Optional) You can change user rights after initial login, as [Users and User Groups](#), on page 38 describes.

Chapter 3: Working with Real-Time Server Settings

When you establish a connection to Real-Time Server, a menu labeled **Settings** appears on the left side (or *sidebar*) of Real-Time Administrator web pages. RT Server Settings listed in this sidebar menu link to pages in which the selected setting's details appear. This chapter explains how to work with RT Server settings generally and briefly defines each link in the Settings menu.

You work with RT Server settings much the same way you work with other settings in the Infor system.

Viewing Server Setting Details

To view the details of any setting, click a link in the **Settings** menu. Details of the setting appear in the main section of the page, to the right of the **Settings** menu.

Changing Server Setting Parameters

To change a parameter for any setting:

- 1 Click a link in the **Settings** menu. Details of the setting appear in the main section of the page, to the right of the **Settings** menu.
- 2 At the bottom of the page, click **Edit**. The field for each parameter you can change becomes active.
Note: Changes to fields identified with the icon take effect only when the relevant RT Server is restarted.
- 3 Enter your settings in the appropriate fields. When you are finished, click **OK** to save your changes or **Cancel** to discard them.

The following table briefly defines each link in the **Settings** menu and indicates the chapter in which it is documented.

Setting Link	Explanation
Server	General RT Server properties. See Server Settings on page 17.

Setting Link	Explanation
Deployment Group	The role of the current RT Server in a deployment group (master, slave, none). See Deployment Groups on page 22.
Port Interface	Real-Time Server ports and authentication. See Port Interface on page 32.
HTTP	HTTP session characteristics. See HTTP on page 31.
Security	Secure Socket Layer setting (enable or disable). See Security on page 35.
Users	Registered users' names, rights, user package group, and authentication. See Users and User Groups on page 38.
User Package Groups	Groups of users configured to use the same default campaign packages (main and alternate). See Users and User Groups on page 38.
User Access Groups	Groups of users who have Marketer rights. See Users and User Groups on page 38.
Data Access	Database services used for lookup, stored procedure, and table sources as well as for recommendation models. See Data Access on page 45.
BIO Access	Profiles for BIO data access from Infor Sales and Service for use in campaign targeting. See BIO Access on page 50.
Java Adapter	Java Virtual Machine (VM) characteristics. See Java Adapter on page 54.
Real-Time Miners	Real-Time miner services (name and description). See Real-Time Miners on page 56.
Recommendation Models	Recommendation model services. See Recommendation Models on page 62.
Campaign Packages	Deployed campaign packages. See Campaign Packages on page 64.
Logging	Log file attributes. See Logging on page 72.

Setting Link	Explanation
Debugging	RT Studio debugger sessions. See Debugging on page 85.
Admin Variables	User-defined information for accessing external web services. See Admin Variables on page 87

The Performance Monitor Displays or logs performance counters characterizing various aspects of RealTime server performance.

See [Performance Monitoring Tools](#) on page 77.

This guide also includes Real-Time troubleshooting information.

See [Troubleshooting](#) on page 82.

Chapter 4: Server Settings

This chapter provides information on server settings, including changing server settings, saving packages, real-time miners, audit trails in the RTDB, restoring lost data and reassigning campaign packages.

Overview

When you click **Server**, the first link in the Settings sidebar menu, details of the current server settings appear in the main page.

Server settings and their defaults are detailed in the following paragraphs:

- **Services Deployment Directory:** Directory that stores deployed campaign packages, and real-time miner resources. By default:

Windows: `C:\Program Files\Epiphany\RT\Services` or

Linux: `<InstallDir>/Services`

This directory includes the followings subdirectories in which Real-Time stores resources used at run time:

- The TargetingEngine subdirectory stores campaign packages.
- The MdapEngine subdirectory stores real-time miners.
- **Enable S&S Single Sign-On:** This option is for integration with Infor Sales and Service. The S&S Single Sign-On (SSO) provides a single mechanism for authentication, access control, and audit common to all Infor products. If your implementation includes a Single Sign-On Server available through Infor Sales and Service, select **Enable S&S Single Sign-On** and enter the appropriate URL in the Single Sign-On Server URL text box.

See the appropriate Sales and Service documentation. Sales and Service is another Infor Enterprise Marketing Suite application. For the latest information about the version of Sales and Service which is supported with Interaction Advisor, see the Platform Support Matrix document (*Platform Support Matrix - Infor Interaction Advisor.doc*) available on the Infor Customer Support web site: <http://www.inforxtreme.com>.

It is advisable to have more than one Single Sign-On server, depending on the kind of failover you want to have. Consider the following options:

- You can have just one Single Sign-On server running on the same machine as Real-Time Server. In this case, if the Single Sign-On server fails, the Real-Time Server also fails. This approach is most efficient in small to mid-sized sites with low volume.

- If you have a deployment group of Real-Time Servers, it is recommended that you have a separate Single Sign-On server for each Real-Time Server. If a Single Sign-On server fails, the accompanying Real-Time Server also fails.
- A third, preferable option is to have two or three machines running Single Sign-On servers and to specify multiple URLs, one on each line, in the Single Sign-On Server URL text box. In this case, Single Sign-On uses the first machine available, jumping to the next machine if the first one fails. This is the recommended architecture for heavy volume sites or sites that require continuous availability.

When SSO is enabled, users must enter the full server DNS for the Real-Time Server when they log on to RT Administrator, Interaction Advisor Manager, and RT Studio. Also see the SSO Note below.

- **Enable AD Single Sign-On.** This option is for integration with Infor Ming.le. If your implementation includes Infor Ming.le, select Enable AD Single Sign-On and AD Single Sign-on Ticket Timeout will default to 30 seconds. This can be increased or decreased, if needed, but the value should always be greater than zero. A value of zero will disable AD Single Sign-On. Also see the SSO Note below.
Note: SSO does not take effect until you restart the server. After enabling SSO and before restarting the Real-Time Server, add at least one user with Administrator rights, or add the default administrative user `rt` to your NTLM domain or directory service. The `rt` user does not work after enabling SSO unless it also exists in your authentication server.
- **Socket Thread Pool Size:** Number of threads available to serve concurrent event requests (as well as requests from Real-Time Administrator and Interaction Advisor Manager) within Real-Time Server. This number must be an integer. The minimum authorized value is 1 and the maximum is 5 times the number of processors (up to a maximum of 128). The default is 5. For most production systems, a good starting point for this setting is between 1.5 and 2 per CPU (depending on relative time spent on calculations and on database IO). Testing and experimenting is the only way to find the optimum value. Above 5 threads of any kind per CPU, server performance can be severely impacted, and random problems such as crawling or hanging may occur.
- **Command Thread Pool Size:** Number of threads available to concurrently execute session wrap-ups within Real-Time Server. This number must be an integer. The minimum authorized value is 1 and the maximum is 5 times the number of processors (up to a maximum of 128). The default is 5, which should be left unchanged if no package using sessions is executed on the server. Otherwise, if the server must perform session wrap-ups, a good starting point for this setting is between 1.5 and 2 per CPU (depending on the amount of database IO during session wrap-ups) for most production systems. Infor recommends that you start with the same setting for Socket Thread Pool Size and Command Thread Pool Size. Testing and experimenting is the only way to find the optimum value. Above 5 threads of any kind per CPU, server performance can be severely impacted, and random problems such as crawling or hanging may occur.
- **Maximum Events Queued:** Maximum number of events received from an event port that Real-Time Server maintains in a to-be-processed queue. When event traffic is moderately heavy, 5 or 10 events may be queued. When the queue is full, Real-Time Server ignores new requests until queued requests have been processed. The suggested value is an integer from 10-500 on Windows and from 100-200 on Linux. The default is 200.
- **Server Socket Read Timeout:** A floating point value indicating the number of seconds Real-Time Server waits for information from an RT client after connecting to a client. This is the maximum time allocated for reading requests from clients, not for processing them. The default is 30 seconds. A value of 0 disables this timeout. This timeout is useful during client debugging for detecting such client errors as not handling incorrect port identification.
- **Client Socket Timeout:** A floating point value indicating the number of seconds an RT client waits after making a Real-Time Server request before presuming an error has occurred. The default is 5

seconds. A value of 0 disables this timeout. This value can be overridden in client applications implemented using the .NET client or the Java Client.

- **Deployment Database Service:** The data access service name of the RTDB for storing real-time miners, campaign packages, and audit trails.

See [Saving Packages, Real-Time Miners, and Audit Trails in the RTDB](#) on page 19 to learn how to enable database storage for these elements.

- **Store Campaign Packages in Database:** Select this option if campaign packages are stored in the identified RTDB. By default this option is not selected.

This option works best when you enable it to update campaign packages in the database and then disable it to ensure that accidental package deletions do not also delete the package from the database. This allows you to retrieve campaign packages from the RTDB if they are accidentally deleted through Real-Time Administrator or Real-Time Studio. Simply re-enable the option to restore the package. Any packages deployed but not saved to the database are also saved when the option is re-enabled.

In the case of upgrade, you will need to ensure that you re-enable this option so that a new backup of your package(s) is made. This is because restoring a package from RTDB which was saved in a prior release is not supported.

- **Store Package Audit Trails in Database:** Select this option if campaign package changes are stored in an audit trail in the identified RTDB and if **Store Campaign Packages in Database** is also set to **Yes**. By default this option is not selected.
- **Store RT Miner Models in Database:** Select this option if real-time miners are stored in the identified RTDB. By default this option is not selected.

Changing Server Settings

See [Server Settings](#) on page 17 for general procedures on changing settings.

Changes to fields identified with the  icon take effect only when the relevant RT Server is restarted.

Saving Packages, Real-Time Miners, and Audit Trails in the RTDB

It is important to configure only one deployment group master or standalone Real-Time Server to store packages, real-time miner information, and audit trails in any one RTDB at the same time.

When you configure a master to store these items in an RTDB, then later change the master to a slave or standalone Real-Time Server, the **Deployment Database Service** list setting changes to (None). If the server has been designated a standalone Real-Time Server, you can select an RTDB data access service name from the list if you like.

To configure a Real-Time Server to store campaign packages, real-time miner data, or audit trails in an RTDB:

- 1 Initialize the RTDB.
- 2 In Real-Time Administrator, connect with the Real-Time Server.
- 3 Click **Data Access** to define a data access service for the RTDB.
See [Data Access](#) on page 45.
- 4 Click **Server** , then click **Edit**.
- 5 In the **Deployment Database Service** drop-down list, select the RTDB data access service name.
- 6 Select **Store Campaign Packages in Database**.
- 7 Select the **Store Package Audit Trails in Database** box to enable audit trail storage in the RTDB.
- 8 Select **Store real-time miner Models in Database** to enable real-time miner storage in the RTDB.
Any campaign packages currently deployed to the Real-Time Server are copied to the RTDB within seconds. Any real-time miner data associated with the campaign package is also automatically stored in the RTDB. Audit trail records are written as soon as any change is made to a campaign package deployed to the current Real-Time Server.

Restoring Lost Data

If a hardware failure occurs that causes the loss of data in the Services Deployment Directory of the Real-Time Server, the data can be restored from RTDB. This data should only be restored from the database when the data was created in the same version of IA into which you are trying to restore.

To restore lost data:

- 1 In Real-Time Administrator, log on to the new Real-Time Server.
See [Starting RT Administrator](#) on page 12.
- 2 Click **Server** and disable RTDB access by deselecting **Store Campaign Packages in Database**, **Store Package Audit Trails in Database**, and **Store RT Miner Models in Database**.
- 3 Click **Campaign Packages**, click **Edit**, and delete all campaign packages and templates deployed to the Real-Time Server.
- 4 Use the appropriate Oracle or SQL Server procedure to restore the RTDB if it was also lost.
- 5 In Real-Time Administrator, click **Server**, click **Edit**, and re-select **Store Campaign Packages in Database**, **Store Package Audit Trails in Database**, and **Store RT Miner Models in Database**.
Within a few seconds, the Real-Time Server begins to synchronize its cached data with the database.

Reassigning Campaign Packages

When a standalone or master Real-Time Server used to store campaign packages in an RTDB must be replaced, use the following procedure.

To reassign storage of campaign packages in the RTDB to another Real-Time Server:

- 1 In Real-Time Administrator, log on to the new Real-Time Server. (See [Starting RT Administrator](#) on page 12.)

- 2 Click **Server**, then click **Edit**, and ensure that **Deployment Database Service** is set to (None) .
- 3 Click **Campaign Packages**, then click **Edit**, and delete all the packages on this Real-Time Server.
- 4 Click **Data Access**, then click **Edit**, then add and create a data access service for the RTDB in which to save campaign packages.
- 5 On the **Server Settings (Edit)** page, select the service name of the RTDB in the **Deployment Database Service** drop-down list and select **Store Campaign Packages in Database**.
- 6 Click **Deployment Group**, then click **Edit**, and set the **Deployment Group Role** to **Master** if the server is intended to be the master of the deployment group.

Chapter 5: Deployment Groups

This chapter defines deployment group settings and Real-Time Server master-slave relationships and then explains how to define and redefine deployment groups. It also explains how to set up back-up servers for failover.

Real-Time Server usually hosts potentially thousands of client requests. When the load exceeds the capacity of a single Real-Time Server host, additional Real-time Servers can be added to the infrastructure to form a deployment group. A deployment group is a collection of one or more Real-Time Servers, running on different hosts, organized in a master-slave structure and managed as a single Real-Time Server System. (If run on same host, there would be resource limitations.) Events distributed to a deployment group are most commonly achieved with an external load balancer.

Deployment Group Support

All Real-Time Servers in a deployment group support the same:

- campaign packages
- underlying services
- data access
- recommendation
- real-time miners
- admin variables

If the master Real-Time Server has either method of Infor Single Sign-On enabled, the slave Real-Time Servers must also have Single Sign-On enabled and must use the same Single Sign-On URL.

See [Server Settings](#) on page 17.

Likewise, slaves must have the Java Adapter enabled if it is enabled on the master. However, slaves need not use the same virtual machine library as the master.

See [Java Adapter Settings](#) on page 54.

Deployment groups simplify campaign administration across multiple Real-Time Servers, including those in different locations. Deployment groups also enable you to share real-time miner models among different Real-Time Servers, as described in [Real-Time Miners](#) on page 56

Deployment groups are implemented using a packet-based protocol on top of TCP/IP. Therefore, the hosts on which all Real-Time Servers in a deployment group reside must be connected using a local area network or wide area network.

Master-Slave Relationships

One Real-Time Server in each deployment group is designated the master server for the group. To make a campaign resource available to a member of a deployment group, you make it available to the master server. The master server automatically distributes deployment information to the other Real-Time Servers in its group, the slave servers, as it becomes available.

If deployment services change when a slave is not available, the master updates the slave's deployed resources when the slave comes online. Slave synchronizations come from their assigned master, and the master verifies that any request for synchronization from a just-come-online slave is of its deployment group. If verification fails, an `unexpected sync request` is logged. When a master updates its slaves's deployment resources, any event a slave is executing completes before the slave starts using the new deployment services.

Only one deployment master or standalone Real-Time Server can be configured to store campaign packages, real-time miner data, or audit trails in a given RTDB at one time.

See [Saving Packages, Real-Time Miners, and Audit Trails in the RTDB](#) on page 19.

Data access services on slaves must not be changed to point to a different server.

Package Master and Marketer rights, described in [Users and User Groups](#), on page 38 are relevant only to the master server in a deployment group. Administrator and Basic rights, however, should be configured for all servers in a deployment group.

Failover Relationships

You can designate one or more backups (or failover) Real-Time Servers for each Real-Time Server. When a principal Real-Time Server is not available, the Real-Time client attempts to use backup servers, in order, if there is more than one. If the backup servers are not available, the request fails.

Note: If you use a hardware load balancer to manage Real-Time Server traffic, you need not designate backup servers. When a Real-Time Server managed by a load balancer becomes unavailable, the load balancer routes events to other servers it manages.

Before logging in, a client can declare its own backup in case its primary server is unavailable. Once login succeeds, the server sends a new list of backups to the client. If the server is the client's primary server, this list replaces any list previously declared by the client. If the client connects to a backup server, it periodically polls its primary and higher ranked backup servers. When a higher ranked server is available, the client sends requests to that server and no longer polls lower ranked servers. For example, when a client is using its third backup, it polls its primary server and first and second backups.

If the first backup becomes available, the client uses the first backup and continues to poll the primary, but no longer polls the second backup. Refer the Real-Time Implementation Guide for information on how to take advantage of failover Real-Time Servers.

RT failover relationships are very flexible:

- The master Real-Time Server can use any of its slaves as a backup.
- A slave can use the master or any other slave as its backup.
- Two slaves can use each other as backups.
- A Real-Time Server can use any other Real-Time Server as a backup whether it is part of a deployment group or not.

Caution: A backup server that is not part of a deployment group is not kept up to date by the primary server and can get out of synchronization with the server it backs up. Therefore, use this configuration with great caution.

Real-Time Servers that participate in failover relationships should use the same user names and password requirements.

See [Users and User Groups](#), on page 38 for more information.

Accessing Deployment Group Settings

You access deployment group settings by clicking **Deployment Group** in the Settings sidebar menu. Information about the current Real-Time Server's deployment group role appears on the Deployment Settings page. This page also includes information for backup servers.

If the current Real-Time Server is not a deployment group member, the following statement appears:

```
Server is not part of a deployment group
```

This does not prevent you from setting up backup Real-Time Servers.

Deployment Group Settings

Deployment Group settings and their defaults are detailed in the following paragraphs for master and slave Real-Time Servers, respectively.

For all Real-Time Servers: if you use SSL, servers can be specified with a fully qualified DNS name or with an IP address. However, using an IP address disables server certificate validation.

Master Real-Time Server Settings

If the Real-Time Server is a deployment group master, deployment group settings are defined as follows:

- **Current Server and Admin Port.** The name or IP address of the current Real-Time Server and its host port number.
- **Server Role.** Master.
- **Group Members.** The identity and host port numbers of all the slave members of the deployment group.
- **Backup Servers and Admin Port.** The identity and host port numbers of the Real-Time Server's backup Real-Time Servers.

Slave Real-Time Server Settings

If the Real-Time Server is a slave, deployment group settings are defined as follows:

- **Current Server and Admin Port.** The name or IP address of the current Real-Time Server and its host port number.
- **Server Role.** Slave.
- **Deployment Master and Admin Port.** The identity and host port number of the master Real-Time Server in the current Real-Time Server's deployment group.
- **Backup Servers and Admin Port.** The identity and host port number of the Real-Time Server's backup Real-Time Server or servers.

Note: All slave servers mimic the master's port assignment so that, for each protocol,

```
port_protocol_on_slave - port_Event_on_slave == port_protocol_on_master  
- port_Event_on_master
```

If you see the following exception after upgrading, use the master's console to change the ports on the slave servers to follow the above rule:

```
Slave server <slave_host> has incorrectly configured  
<event, admin, or other> port. It is <n1> but should be <n2>.
```

The port on the slave can be zero only if it is also zero on the master.

Defining a Deployment Group

- 1 Defining a deployment group is a three-part process consisting of the following general steps:
 - a Designating a master Real-Time Server
 - b Designating slave Real-Time Servers
 - c Associating the slaves with the master

2 To designate a master Real-Time Server:

- a Start Real-Time Administrator on the Real-Time Server you want to designate as master.
- b Click **Deployment Group** in the Settings sidebar menu.
- c On the **Deployment settings** page, click **Edit**.
- d In the **Deployment Group Role** pull-down list, select **Master**.

Note the master's Current Server and Admin Port values. (You will use these values when you define slaves.)

Note: Each time you refer to a Real-Time Server other than the Real-Time Server you are connected with, you must use the values that appear in the Current Server fields of the Deployment settings page for the other Real-Time Server.

- e (Optional) Add backup servers as described in [Designating Backup Servers](#) on page 29.
- f Click **OK** to save your changes.

3 To designate slave Real-Time Servers:

- a Start Real-Time Administrator on a Real-Time Server you want to designate as a slave.
- b Click **Deployment Group** in the Settings sidebar menu.
- c On the Deployment settings page, click **Edit**.
- d In the **Deployment Group Role** pull-down list, select **Slave**.

Note the values of the **Current Server** and **Admin Port**. You will use these values later when you associate the slave with the master.

Note: Each time you refer to a Real-Time Server other than the Real-Time Server you are connected with, you must use the values that appear in the Current Server fields of the Deployment settings page for the other Real-Time Server.

- e (Optional) Add backup servers as described in [Designating Backup Servers](#) on page 29.
- f In the **Deployment Master** and **Admin Port** fields, identify the master Real-Time Server, using the master's **Current Server** values. (See "step e" of the preceding procedure.)
- g Repeat this procedure for each Real-Time Server you want to designate as a slave.

4 To associate the slave Real-Time Servers with the master Real-Time Server:

- a Start Real-Time Administrator on the master Real-Time Server.
- b Click **Deployment Group** in the Settings sidebar menu.
- c On the **Deployment settings** page, click **Edit**.
- d In the bottom half of the page, enter the **Server** and **Admin Port** values for the slave Real-Time Server you want to associate with the master.
- e Click **Add**. The Real-Time Server you added appears listed under **Group Members**.
- f Repeat [Step d](#) on page 26 and [Step e](#) on page 26 for each slave Real-Time Server you want to add.
- g Click **OK**.

Redefining a Deployment Group

When a deployment group master Real-Time Server is configured to save campaign packages, real-time miners, or audit trails in an RTDB, reassigning the role of master to a different Real-Time Server automatically configures the new master to write to the RTDB.

Adding a Slave to an Existing Deployment Group

To add a slave to an existing deployment group:

- 1 Connect with the Real-Time Server you want to add.
- 2 Designate this Real-Time Server as a slave.
- 3 Connect with the master Real-Time Server.
- 4 Add the slave Real-Time Server to the deployment group.

Adding a Slave Containing Real-Time Miner Data

Adding a Real-Time Server that contains real-time miner data to a deployment group deletes the new slave's real-time miner data.

Note: In both of the scenarios described in this section, the new slave's real-time miner data is deleted, even if you remove a slave and immediately add it again.

Deployment Group Shares Real-Time Miner Learnings

If you add a Real-Time Server that has real-time miner learnings on it to a deployment group that shares real-time miner learnings within the deployment group (where the master has **Store RT Miner Models in Database**: enabled), the new slave's real-time miner data will be deleted and the newly added slave will receive the real-time miner data from the RTDB.

This occurs because, in a deployment group, the campaign packages are pushed from the master to the slaves. The real-time miner on each Real-Time Server must be correct for the campaign packages being used. Therefore, the real-time miner definition from the master is also pushed to the slave. Since the data in the master must be consistent with the service definition and with the campaign package, the slave's version of the real-time miner is discarded and replaced with the data from RTDB. In other words, the master's version of the real-time miner is the only trusted source. The real-time miner data on the slave is discarded.

Deployment Group Does Not Share Real-Time Miner Learnings

If you have a Real-Time Server that has real-time miner learnings on it and you add it to a deployment group that does not share real-time miner learnings within the deployment group (where the master has **Store RT Miner Models in Database: DISABLED**), the new slave's real-time miner data will be deleted and it will not receive real-time miner data from the RTDB. Therefore, the slave will start its real-time miner learning fresh.

This occurs because, in a deployment group, the campaign packages are pushed from the master to the slaves. The real-time miner on each Real-Time Server must be correct for the campaign packages being used. For instance, the set of fields, offers, and levels should follow the campaign package deployed on the server. To ensure consistency, the service definition from the master is copied to the slave. However, the content of the real-time miner, which must also follow the master's copy, cannot be populated since the deployment group does not store its real-time miners in the RTDB. As a consequence, the only way to guarantee the slave real-time miner's consistency is to wipe the data out and start fresh.

Assigning a Slave the Role of Master

If the master Real-Time Server becomes temporarily unavailable (for example, in case of hard disk failure), you can reassign the role of master to another Real-Time Server in a deployment group. Any slave in the deployment group can be reassigned as master, but will only have the latest deployed package(s). The slaves are not kept updated for changes in the packages being edited. They are only up to date for the latest Deployed changes.

To reassign the role of master to another Real-Time Server:

- 1 Start Real-Time Administrator on the slave Real-Time Server you want to designate as master.
- 2 On the **Deployment settings** page, change this slave's **Deployment Group Role** to **Master**.
- 3 Remove the former slave from the new master's **Group Members** list.
- 4 Start Real-Time Administrator on each slave and identify the new master.
- 5 Configure the new master, if you like, to store campaign packages, real-time miners, or audit trails in the RTDB that was used by the former master.
- 6 When the former master becomes available again, change one of the two masters to play the role of a slave, as follows:
 - If packages, miners, or audit trails are being stored in an RTDB, you can demote either the current master or the former master to a slave.
 - If these items are not being stored in an RTDB, demote the former master to a slave until its packages have been resynchronized with those of the current master.

Splitting a Deployment Group

When you split a deployment group into two groups, first configure the slave to serve as the master of the new group and reconfigure it as a standalone Real-Time Server, then reconfigure it again as a master.

To split a deployment group into two groups:

- 1 Start Real-Time Administrator on the slave Real-Time Server you want to designate as master.
- 2 On the **Deployment settings** page, change this slave's **Deployment Group Role** to **Server does not participate**.

This step automatically clears the **Deployment Database Service** on the **Server settings** page, which prevents the new master from attempting to save campaign packages, miners, or audit trails to the same RTDB as the original master.

- 3 Configure the standalone Real-Time Server as master.
- 4 To have the new master save packages, miners, or audit trails in an RTDB, set up a data access service identifying a different RTDB from the one used by the original master.
- 5 On the **Server settings** page, identify the service in the **Deployment Database Service** list.

Removing a Slave

To remove a slave from a deployment group:

- 1 Start Real-Time Administrator on the master Real-Time Server.
- 2 Click **Deployment Group** in the **Settings** sidebar menu.
- 3 On the **Deployment settings** page, click **Edit**.
- 4 In the **Group Members** list, select the slave you want to remove and click **Delete**.
- 5 Start Real-Time Administrator on the slave Real-Time Server you want to remove from the group.
- 6 On the **Deployment settings** page, click **Edit**, then select **Server does not participate** in the **Deployment Group Role** pull-down list.

Designating Backup Servers

A Real-Time Server can use any other designated Real-Time Server as a backup, whether you are using deployment groups or not. When a principal Real-Time Server is not available, the RT client tries the backup server. If there is more than one backup server, the client tries them in order until it finds one that is available. If none are available, the request fails.

To designate backup servers:

- 1 Start Real-Time Administrator on the Real-Time Server you want to designate as master.
- 2 Click **Deployment Group** in the **Settings** sidebar menu.

- 3 On the **Deployment settings** page, click **Edit**.

Note: Each time you refer to a Real-Time Server other than the Real-Time Server you are connected with, you must use the values that appear in the **Current Server** fields of the **Deployment settings** page for the other Real-Time Server.

- 4 In the **Backup Servers** and **Admin Port** fields, designate another Real-Time Server to serve as the master's backup resource.

Ensure that the same user names and password requirements have been configured for the master server and its backup server.

See [Users and User Groups](#) on page 38.

- 5 Click **Add**.

- 6 Repeat [Step 4](#) on page 30 and [Step 5](#) on page 30 as desired until all desired backup servers are entered. If the servers you enter here have their own backups, events will not fail over to them.

- 7 To adjust the order of the servers in the **Backup Servers** list:

- a Select a server in the list.
- b Click **Up** or **Down** to change the order.

- 8 Click **OK** to save your changes.

Chapter 6: HTTP

HTTP settings govern communication between Real-Time Server and the following:

- Clients implemented using Web/ HTTP, Java or .NET
- Interaction Advisor Manager
- Real-Time Administrator

You access HTTP settings by clicking **HTTP** in the Settings sidebar menu. The current HTTP settings appear on the HTTP Settings page.

Changes you make take effect only when the RT Server you are working with restarts.

HTTP Settings

HTTP settings and their defaults are detailed as follows:

- **HTTP Content Base Directory.** The directory in which Real-Time Server command files used by the clients reside. By default:
 - Windows: `C:\Program Files\Epiphany\RT\HTTPContent` or
 - Linux: `<InstallDir>/HTTPContent`
- **Enable HTTP Logging.** Indicates whether HTTP requests and header information is logged. By default, not selected.
- HTTP Logging is useful for debugging Real-Time Server interactions that use the HTTP protocol. However, the log file size increases rapidly, so HTTP logging should be enabled only when necessary. Logging also slows down HTTP request processing.
- **HTTP Log File.** The HTTP log file name and location. By default:
 - Windows: `C:\Program Files\Epiphany\RT\LogFiles\httplog.txt`
 - Linux: `<InstallDir>/LogFiles/httplog.txt`

Chapter 7: Port Interface

Infor Real-Time Server can use as many as seven ports to process requests, three administrative ports and four event ports. The administrative ports are intended for use with Interaction Advisor utilities, such as RT Studio and Interaction Advisor Manager. The event ports are for use by client applications. You can change the port assignments in Real-Time Administrator. You can also configure port level timeout and user authentication criteria.

You access Port Interface settings by clicking **Port Interface** in the **Settings** sidebar menu. The current configuration of each port appears on the **Port Interface Settings** page.

Port Interface Settings

Port Interface settings and the defaults are described here.

- 1 The port interface settings and the defaults are.

Port Name

One of the available port names

Value

Port number currently assigned. If another application is using the default ports, specify a number below 65535 that is not in use. Infor suggests you select a number higher than 5001 to minimize the likelihood of conflicting with other applications.

To disable a port interface, set the value to 0. If you are using SSL, you may want to disable all non-SSL ports to enhance security.

Timeout

Also referred to as Service Guarantee Timeout, this is the number of seconds after which an unserved request from the port is dropped. This number represents the maximum time Real-Time Server is expected to require to execute a request arriving on a particular port, including time waiting in the to-be-processed queue. The to-be-processed queue value is set on the **Server settings** page (**Maximum Events Queued**). You can specify a floating point value

Generally, admin port timeouts should be longer than event port timeouts because administrative requests (such as package deployment) can be time consuming.

Note: The parameters following this note are not available when either method of Single Sign-On is enabled.

Password Authentication

Indicates the password authentication scheme used by default for requests received using the port.

- **None** indicates passwords are not checked.
- **Operating System** indicates that a request must be accompanied by a valid operating system password.

You can override default password authentication for individual users who are registered with Real-Time Server on the **Users Settings** page. The default password authentication scheme is always used for unregistered and anonymous users.

Anonymous Users Allowed

Indicates whether the port supports requests from users who provide no user name. The default is **Yes**.

Anonymous User Rights

Indicates whether or not anonymous users of the port have Basic rights. Users with Basic rights are entitled to run RT client applications that use the port. To remove Basic rights, select **None**. The default value is **Basic**.

- **Administrator** rights enables users to use all RT utilities and client applications.
- **Package Master** rights give users complete control of all campaign packages deployed to the Real-Time Server you are working with.
- **Marketer** rights are primarily for individuals who define and manage particular RT campaigns, such as marketers and business managers.
- **Basic** rights give users the right to run client applications using the port.
- **None** denies all rights to anonymous users, making it impossible for them to use any RT applications. For more information on rights, See [Users and User Groups](#) on page 38

2 The following table lists default Real-Time Server port assignments.

Type	Name	Default	SSL Secure	Use From
Admin	Admin	7200	Either ^a	RT Studio
	Admin_HTTP	7201	No	Interaction Advisor Manager Real-Time Administrator
	Admin_HTTPS	7203	Yes	Interaction Advisor Manager Real-Time Administrator
Admin	Event	7205	No	RT Studio test facility

Type	Name	Default	SSL Secure	Use From
	Event_HTTP	7206	No	Java Clients Socket Component clients Web/HTTP Component clients .NET clients
	Event_SSL	7207	Yes	RT Studio test facility
	Event_HTTPS	7208	Yes	Java Clients Socket Component clients Web/HTTP Component clients .NET clients

May be secure or non-secure depending on the Admin Port Uses SSL Protocol setting on the Security page.

See [Security](#) on page 35 for more information.

Changing Port Interface Settings

Changes you make take effect only when the RT Server you are working with is restarted.

To change a parameter for any Port Interface setting:

- 1 Click **Port Interfaces** in the **Settings** menu. Details of the setting appear in the main section of the page, to the right of the **Settings** menu.
- 2 At the bottom of the page, click **Edit**. The field for each parameter you can change becomes active.
- 3 Click **modify** to access settings for the port you want to modify.
- 4 Enter your settings in the appropriate fields.
- 5 When you are finished, click **OK** to save your changes.

Chapter 8: Security

This chapter explains the Secure Socket Layer security option and security module certificates. Infor Interaction Advisor incorporates the Secure Sockets Layer (SSL) protocol to encrypt and authenticate communications across the open Internet. When SSL is installed, you can use it when sending messages:

- Between Real-Time Server and Real-Time Administrator. SSL is used when you select **Use Secure Port** on the start page of Real-Time Administrator.
- Between Real-Time Server and Real-Time Studio when you select **Secure Connection Option** in Real-Time Studio.
- Between Real-Time Server and client applications that use the Java or .NET clients. SSL is used when the properties provided by these interfaces are set to **true**.
- Between Real-Time Servers in a deployment group when the **Admin Port Uses SSL Protocol** option is selected.

When you use SSL to communicate with Real-Time Server, messages, including user names and passwords, are encrypted to safeguard their content.

The Security Module

To configure Real-Time Server security options, click **Security** in the Real-Time Administrator **Settings** sidebar menu. The **Security Settings** page indicates whether the Security module is enabled and whether the Admin port uses SSL protocol.

To enable or disable SSL communications:

- 1 In the **Settings** sidebar menu, click **Security**, then click **Edit**.
- 2 Select the options you want to enable, and click **OK**. Changes take effect the next time Real-Time Server starts.

Security Certificates

You must obtain a certificate for Real-Time Server to ensure that the Interaction Advisor client applications consider the RTServer as a trusted source. The certificate is obtained from a trusted,

third-party organization-- the Certificate Authority. The Certificate Authority requires detailed identification information before issuing a certificate.

Interaction Advisor client applications, implemented using the Java and .NET clients require a certificate issued by a X.509 compliant certificate authority.

Note: Infor supports any certificate issued by X.509 compliant certificate authority. Free trial certificates can be used for testing, but are not recommended for production.

By default, Real-Time Server's Security module is enabled and has a signed certificate installed. This certificate is not valid for your server and is for test purposes only. Before installing your signed certificate, it is recommended that you copy the server.pem file to back up the signed test certificate. (You can copy this server.pem file back, in case an error occurs after the installation of the new signed certificate).

The test certificate is not signed by a trusted authority; only the web browser allows the user to bypass the non-trusted certificate. As a result, the limitations of these certificates are:

- You cannot use the secure connection in Real-Time Studio.
- You cannot run events to Real-Time Server except for events that are run in a web browser.
- You can access the Real-Time Server using the secure connection from Interaction Advisor Manager or Real-Time Administrator, but a security alert is displayed.

Obtaining a server certificate

To obtain a server certificate:

- 1 On the **Security** settings page, click **Make a New Server Certificate Request** to specify:

Country Code

Specify the ISO country code. For example, select **us** for the United States.

State or Province

Specify the state or the province. Do not use abbreviations.

Locality or City

Specify the name of the city or locality.

Organization Name

Specify the legal name of your company. You must specify the registered owner of the domain as specified in the **Domain** field. Avoid abbreviations except for minor terms such as Incorporated (Inc.) or Corporation (Corp.).

Organizational Unit

Specify the name of the department or division responsible for the administration of the Real-Time Server.

Domain

Specify the fully qualified domain name of the Real-Time Server host secured by the certificate. Specify exactly what the users specify to access the secure site, for example, from Real-Time Administrator or Interaction Advisor Manager. Do not include directories.

E-mail

Specify an e-mail address.

Message Digest

Select the encoding algorithm for the signed certificate. Supports secure hash algorithms of SHA-1 and SHA-256. Additional information on Hash Functions is available online in the National Institute of Standards and Technology (NIST) Policy.

Size of the key

Select the required key size in bits. Higher key size is more secure. However, this requires additional computation time.

- 2 To generate the encrypted private key, click **OK**. The encrypted private key is generated and the file is saved as `private.pem`.
- 3 Submit the content of the Server Certificate Request (also known as a Certificate Signing Request or CSR), to the Certificate Authority. When the Certificate Authority requests for Server Platform information, select **Apache**.
- 4 After you receive the Signed Certificate from the Certificate Authority, read the provider's accompanying instructions, then on the **Security** settings page, click **Install a Signed Server Certificate**, and copy the certificate in the text box.

Note: You can also install 5 Intermediate certificates on the page along with the signed certificate. To install these intermediate certificates, click **Add** another intermediate certificate. The instructions from the Certificate Authority will indicate if additional certificates will need to be installed. The order in which you add intermediate certificates is critical. It is recommended that the intermediate certificate listed closest to the signed certificate in the instructions from your Certificate Authority is installed next, and so on.
- 5 To install and verify the certificate, click **OK**. The previously created private key and the signed certificate replace the contents of `server.pem`.
- 6 Restart Real-Time Server for the certificate to come into effect.
- 7 For any clients that will be using the secure ports, you may also need to ensure the root certificate is updated as per the instructions from your Certificate Authority. For browser clients, this can be done via your browser menu options. For clients such as .NET or Java, you will need to update the `rootcerts.pem` file. (For Java, this file is inside the `rpbean.jar` file.)

Changing Certificates

You need not uninstall or move the existing certificate to change to a new one. Real-Time Server automatically renames the old certificate file when you change to the new one.

Chapter 9: Users and User Groups

This chapter explains how to configure settings for users, user access groups, and user package groups.

Users

Rights define the applications an individual can use to interact with a particular Real-Time Server. Together with permissions, rights determine a user's Real-Time privileges.

This section provides an overview of privileges, then describes how to use the **Users Settings** page to work with user rights. To work with rights for anonymous users, for whom no user name is known to Real-Time Server, See [Port Interface](#) on page 32

If either method of Single Sign-On (SSO) is enabled, anonymous users are not allowed.

Privileges

Users' privileges to perform Real-Time activities are governed by their rights and permissions. Rights are Real-Time Server-level privileges that define the applications an individual can use to interact with a particular Real-Time Server. The term permissions denotes privileges at the campaign package, campaign, and offer levels. Permissions regulate changes users with Marketer rights can make to a particular campaign package using Interaction Advisor Manager.

Rights

Real-Time Server-level rights are granted and managed on the **Users settings** and **Port Interface settings** pages in Real-Time Administrator. Each level of rights includes the rights of all lower levels. From most to least powerful, rights include these levels:

- **Administrator**
- **Package Master**
- **Marketer** rights
- **Basic** rights

Permissions

Permissions are privileges users with Marketer rights must have to make changes to a particular campaign package using Interaction Advisor Manager. Administrator and Package Master rights include permission to make such changes. Marketers must be granted specific permission in order to change or deploy campaign packages. Permissions can be defined at three levels:

- Package
- Campaign
- Offer

At each of these three levels, you can set up Owner, Editor, and Reviewer permissions. Owner permissions entitle you to define Editor permissions. Editor permissions enable users with Marketer rights to modify packages, campaigns, and offers using Interaction Advisor Manager.

Owner permission can be granted by an authorized user or automatically when a campaign or offer is created. When created, campaigns and offers are assigned, by default, the current Editor permissions for the package. You can redefine campaign and offer Editor permissions.

At the package level, you can also define Package Deployer permissions. Package Deployers are the only Marketers who can deploy production version campaign packages. There are no Reviewer permissions at the package level.

Only one individual at a time can own a package, a campaign, or an offer. But multiple individuals can hold any of the other permissions. You can grant permissions either to individual users or to user access groups, collections of users to whom Marketer rights have been granted. Like rights, user access groups are defined using Real-Time Administrator.

See [User Access Groups](#) on page 42.

User Settings

The Users settings page lists information for existing users

User

The user name registered with Real-Time Server.

If either method of SSO is enabled, the user name must be included in your authentication directory. The user name may belong to either an individual or a role that includes people who perform the same work. Real-Time user access groups are not available with SSO enabled. If you add a role, the users who have that role all have the same rights when they log on. If you are using SSO, you should work closely with your network or authentication directory administrators.

If S&S SSO is enabled, the list at the bottom of the User Settings page can be used to select users or groups from your SSO authentication service or type the user name directly in the **User** field.

The names in the list are fully-qualified as follows:

- NTLM user IDs are typically: MYCOMPANY\userID
- LDAP user IDs are typically:
uid=userid,ou=people,dc=mycompany,dc=com
- ADS user IDs are typically:
cn=userid,cn=users,dc=mycompany,dc=com

Rights

The rights assigned to the user. Each level of rights includes the privileges available at lower levels:

- **Administrator** rights entitle the user to use all Real-Time utilities and client applications.
- **Package Master** rights grant the user complete control of all campaign packages deployed to a particular Real-Time Server. Package Masters can create and deploy packages. Package Masters can also deploy or discard other users' changes to a campaign package and grant users with marketer rights permission to change campaign elements.
- **Marketer** rights are primarily intended to enable such users as marketers and business managers to define and manage particular RT campaigns. In Studio, Marketers can always perform all offer and campaign level activities, but no package level activities. In Manager, Marketers are governed by Permissions for offer, campaign, and package level activities.
- **Basic** rights entitle the user to run client applications using the port.
- **None** denies all rights, making it impossible for the user to use any RT applications.

User Package Group

The user package group to which a user has been assigned using the **User Package Groups** settings page.

Authentication

An indication of the password authentication scheme used by default for requests received from the user. When AD Single Sign-On is enabled, this setting should be set to 'None.' This setting does not apply when S&S Single Sign-On is used.

- **None** indicates the password is not checked.
- **Operating System** indicates that a request must be accompanied by a valid operating system password.
- **Port Default** indicates password authentication specified for the port on which a request is received should be used. Port password authentication is defined using the **Port Interface settings** page.

Select from the List

Select users or groups from this list when S&S SSO is enabled to avoid typing errors. Click **Show Individual Users** to include individuals in the list.

Note: Ensure that you set up the same user definitions for Real-Time Servers that participate in failover relationships.

See [Deployment Groups](#) on page 22.

Also ensure that when assigning **Marketer** and **Package Master** rights, you are connected with either a standalone Real-Time Server or a Real-Time Server that is the master in a deployment group.

Defining Users

You access Users settings by clicking **Users** in the **Settings** sidebar menu. You register users with Real-Time Server and configure rights, package groups, and password authentication criteria for these users on the **Users settings** page. The **Users settings** page looks different when S&S SSO is enabled. Review the [User Settings](#) on page 39 for additional information.

To add a registered user:

- 1 On the **Users settings** page, click **Edit**.
- 2 On the **User settings (Editing)** page, click **add**.
- 3 In the **User Name** field, specify a user name. Or, if S&S SSO is enabled, you may select a user or role from the list at the bottom of the page.

Case sensitivity of the user value you specify in the **User Name** field:

- Windows: User names are case insensitive during creation and also when used to log in. For example, if you create a user name **USER1**, you cannot create **user1**. However, you can use **USER1** or **user1** to log in.
- Linux: User names are case insensitive during creation, but case sensitive when used to log in. Hence, the case of the user name must match with the one used during creation. For example, if you create **USER1**, you cannot create **user1**. To login, you, must use **USER1**; **user1** is not valid.

Note: The value can only contain alphanumeric characters or these symbols: . - _ , = \. In addition, the value may contain a single space. If any trailing or leading spaces exist at start or end of value, the system will trim them before saving the value. If there are any occurrences of multiple adjacent spaces inside the value, the system will convert these to one space before saving the value. When logging into RT Administrator, IA Manager, or RT Studio, trimming does not occur on the value supplied for the **User Name** field for login. The user must provide a value that matches a value in the Users list This value must not have a leading or trailing space, and only one space is permitted in the value string.

- 4 In the **Rights** pull-down list, assign rights to the user or role.
See [Privileges](#) on page 38.
- 5 In the **User Package Group** pull-down list, select a package.
- 6 If you are not using SSO, in the **Password Authentication** pull-down list, select a password checking scheme for the user. **Default** refers to the authentication specified for any port on which the user's requests are received using the **Port Interface settings** page. If you are using S&S SSO, this list does not appear.
- 7 Click **OK** to save your user definition. The user is immediately available for use.
- 8 Click **Done** when you are finished working with users.

To Modify User Information

- 1 On the **Users settings** page, click **Edit**, then click **modify** in the same row as the user you want to modify. The **Users settings - Modify a User** Page appears. You can change Rights, User Package Group, and Password Authentication (if available) in the appropriate drop-down lists.
- 2 Make your changes, then click **OK** to save them. Your changes take effect the next time the user submits a request, except when you give users with Basic rights new, higher rights. In order to raise rights from Basic, you must restart the Real-Time Server.

To Delete a User

- 1 On the Users settings page click **Edit**, then click **delete** in the same row as the user you want to remove.

- 2 Click **OK** to confirm the deletion.
- 3 The user name you deleted is immediately unavailable to process requests with the current Real-Time Server.

User Access Groups

A user access group is a collection of registered users who have been assigned Marketer rights using the Users settings page. User access groups are convenient when the same permissions must be assigned to multiple Marketers in RT Studio or Interaction Advisor Manager.

To access User Access Groups settings, click **User Access Groups** in the Settings sidebar menu.

Note: User access group information is managed within Real-Time and does not apply to either method of Single Sign-on because responsibility for user access group information is delegated to an external directory.

User Access Group Settings

User Access Group settings are defined as follows:

User Access Group Name

The name assigned to the group.

Members

By default, names of all registered users with Marketer rights appear in the **No** box.

User Package Groups

A user package group is a collection of users configured to use the same campaign package by default when an Real-Time client sends an event that has no campaign package name associated with it.

Using groups facilitates testing new packages. Assign certain users to groups you want to act as test groups for new or changed campaign packages. The test groups use a campaign package, usually a test version of the package, until you consider the package ready for widespread use. When the package is deployed as a production version, the production group of users can use it.

User Package Group Settings

When you click **User Package Groups**, the **User Package Groups** settings page displays information for existing user groups:

User Package Group Name

A user defined name for the user group.

Main Package and Alternate Package

The group's designated main and alternate default campaign packages. The main package is used when the package named in an event does not exist. If the main package is also non-existent, the alternate package is used. For example, if a group is configured to be used by default a main package named `Golden_Test` and an alternate package named `Golden`, any users assigned to this group automatically use `Golden_Test` whenever `Golden_Test` is available, but use `Golden` when `Golden_Test` is not available.

Procedures

To define a user package group:

- 1 Click **User Package Groups** in the Settings sidebar menu.
- 2 Click **Edit** on the User Package Groups Settings page.
- 3 In the **User Group Name** column, click **add**.
- 4 In the **Group Name** text box, enter a name for the new user package group.
- 5 In the **Main Package** drop-down list, select a package. Packages listed here are deployed to the current Real-Time Server.
- 6 In the **Alternate Package** drop-down list, select a package.
- 7 Click **OK**.
- 8 Add registered users to the user package group. See [Defining Users](#) on page 40.
- 9 The user's group affiliation takes effect the next time a package group's user connects with the current Real-Time Server.

Modifying a default package

To modify default package assignments:

- 1 Click **User Package Groups** in the Settings sidebar menu.
- 2 Click **Edit** on the User Package Groups settings page, then click **modify** in the same row as the group you want to work with or next to the **Server wide default package** field.
- 3 Make your changes, then click **OK**. The changes take effect immediately.

Removing a user group definition

To remove a user group definition:

- 1 Click **User Package Groups** in the Settings sidebar menu.
- 2 Click **Edit** on the User Package Groups settings page, then click **delete** next to the group you want to remove.

- 3 Click **OK** to confirm the deletion. The user group is no longer available.

Chapter 10: Data Access

Real-Time Server must be configured to access databases in which resources reside for lookups, stored procedures, tables, recommendation models, Data Mart profiles, real-time miners, and campaign packages. For each such database, including the RTDB, a data access service must be defined.

All RTDB data access services must point to the same RTDB database. The data access service for the RTDB you use to store campaign packages or real-time miners must be identified as the Deployment Database Service for only one Real-Time Server, either a deployment group master or a standalone Real-Time Server. Data access services on slaves must not be changed to point to a different server.

You reach Data Access settings by clicking **Data Access** in the Settings sidebar menu. Current settings appear on the Data Access page.

This includes the data access services on slaves in a deployment group. They must all point to the same RTDB server.

Database Access Rights

Database access rights are required to support the definition and use of campaign elements based on database tables, views, and stored procedures. When you define a data access service for a database, ensure that the user associated with the service has the database privileges listed in the following table:

To Use a	The Data Access Service User Name Must Have		
	SQL Server	DB2	Oracle
Lookup source	db_datareader	CONNECT privilege for the target database SELECT privilege for the target table	CREATE SESSION privilege for the target database SELECT privilege for the target table
Table source	db_datareader db_datawriter	CONNECT privilege for the target database SELECT, UPDATE, DELETE, and INSERT privileges for the target table	CREATE SESSION privilege for the target database SELECT, UPDATE, DELETE, and INSERT privileges for the target table

To Use a	The Data Access Service User Name Must Have		
	SQL Server	DB2	Oracle
Stored procedure source	db_ddladmin db_datareader db_datawriter	CONNECT privilege for the target database EXECUTE privilege for the target stored procedure	CREATE SESSION privilege for the target database EXECUTE privilege for the target stored procedure

Data Access Settings

Data Access settings and their defaults are detailed in the following paragraphs.

Logical Name

The name used when defining lookup sources, tables, Data Mart profiles, and other campaign objects using RT Studio. Logical names must be unique to the Real-Time Server; case does not matter.

Note: When creating a Data Access service for the RTDB, a value of **RTDB** is suggested for **Logical Name**. If a different value is defined, then you will need to modify the `AppLib_Options.js` file in 4 places by changing the references from RTDB to the customized name. The `AppLib_Options.js` file is located at `<RT Installation Directory>/AppLib/AppLib_Options.js`. See the JavaScript Reference Online Help in RT Studio for more information about the Application Library which is utilized by packages created from Power Templates.

Data Source Type

ODBC, DB2, or Oracle. If Real-Time Server is installed as a Windows service, ODBC sources must be defined as system DSNs.

Source

A name identifying the database. This is a system data source name for ODBC, a CLI data source name for DB2, or a server instance name from `TNSNAMES.ORA` for Oracle.

Database name

Optional. Visible only when ODBC has been selected as the Data Source Type. It specifies a database name to override the default login database name or the login database name specified in the ODBC Data Source Name setup. The database name is case sensitive if the database server was installed as case sensitive or the database was created with case-sensitive collation.

User Name

The database user identifier used to establish a database connection. The users you specify must have adequate database privileges to support any lookup, table, or stored procedure sources used by campaign packages deployed to the current Real-Time Server.

Note: Blanks should not be used in this value.

Password

Database password for the user.

Note: Blanks should not be used in this value.

Multi Subnet Failover

Optional. Visible only when ODBC has been selected as the **Data Source Type**. When this check box is selected, the ODBC access library submits `multiSubnetFailover=yes` as a connection string attribute, at runtime. This enables faster failover connectivity when clustered nodes are in a single subnet or multiple subnets. However, this check box must be selected only if this setting is supported by the SQL Server configuration.

Trim trailing blanks for fixed length character data types

By default, this setting is unselected. When this setting is selected, trailing spaces are removed as follows:

- CHAR, NCHAR columns from a lookup
- CHAR, NCHAR output parameters from a stored procedure
- CHAR, NCHAR result columns from a stored procedure

Minimum Connections

Number of database connections Real-Time Server should obtain at startup. The default minimum is zero. The recommended value is 1. Infor recommends setting minimum connections to 1 for all DB2 type data access services.

Maximum Connections

Maximum number of database connections Real-Time Server is allowed to create for the given data source. The default is 5. The number of connections determines the number of data access requests that can be processed in parallel. The optimum number depends on database server capacity. It should be at least as many as socket or command threads but it should not exceed the combined number of socket and command threads. For most production systems the optimum number is between 1.5 and 2.5 per CPU (1.25 times the socket or command thread pool size is a good estimate). Above 3 per CPU, server performance is adversely impacted, and random problems such as crashing or hanging may occur.

See [Server Settings](#) on page 17.

OfferTracker

Most client packages have an OfferTracker table that is mapped to a Data Access service. This table to which the RpOfferTracker view is connected is provided in the RTDB, delivered as part of the product. The table stores communication history of extensions and acceptances. There are many other tables and stored procedures in the RTDB to ensure that the communication history is recorded properly. Preserving the integrity of this data is important. Hence, these procedures must be adhered to whenever you make updates to redirect the OfferTracker table to a new location:

- 1 Stop sending events to RT Server.
- 2 Make the changes. For example:
 - a Change the Data Access service selection for the OfferTracker table in the package using RT Studio.
 - b Update the Data Access service used by OfferTracker in RT Administrator.
 - c Change the database configuration.
- 3 Stop and Restart RT Server. This ensures that the Ids related to OfferTracker tables are synchronized between the package service and the database.

- 4 Resume sending events to RT Server.

Deleting a Data Access Service

To delete a data access service:

- 1 On the Settings sidebar menu, click **Data Access**.
- 2 Click **Edit** on the **Data Access settings** page, then click **delete** in the same row as the service you want to remove.
- 3 Click **OK** to confirm the deletion. The data access service is no longer available.

Adding a DB2 Data Access Service

To add a DB2 data access service:

- 1 On the Settings sidebar menu, click **Data Access**.
- 2 On the Data Access settings page, click **Edit**, then click **add**.
- 3 In the **Logical Name** field, enter a unique name for the Real-Time Server.

Data Source Type

Select **DB2**.

Source

A DB2 CLI data source name. On Linux, the CLI data source name is listed in the `db2cli.ini` file associated with the `$DB2INSTANCE` home.

User Name

The DB2 database user ID.

Password

The DB2 database password.

Minimum Connections

The number of database connections Real-Time Server must obtain at startup.

Maximum Connections

The maximum number of database connections Real-Time Server must cache.

Description

Optional. A description for the database service

- 4 Click **OK**. The data access service you added is immediately available.

Adding an Oracle Data Access Service

To add an Oracle data access service:

- 1 On the **Settings** sidebar menu, click **Data Access**.
- 2 On the **Data Access settings** page, click **Edit**, then click **add**.
- 3 Specify this information: In the **Logical Name** field, enter a service name.

Data Source Type

Select **ORACLE**.

Source

The SQL *Net connect string used to access the server on which the Oracle database reside.

User Name

The Oracle database user ID.

If the Data Access Service is for the RTDB and the user is not the RTDB creator, the user must have the Oracle RTUSER role. The RTADMIN role allows a user to execute RTDB stored procedures intended for administrators.

Password

The Oracle database password.

Minimum Connections

The number of database connections Real-Time Server must obtain at startup.

Maximum Connections

The maximum number of database connections Real-Time Server must cache.

Description

Optional. A description for the database service

- 4 Click **OK**. The data access service you added is immediately available.

Modifying a Data Access Service

To modify a data access service:

- 1 On the Settings sidebar menu, click **Data Access**.
- 2 Click **Edit** on the **Data Access settings** page, then click **modify** in the same row as the service you want to change.
- 3 Make your changes, then click **OK**. Your changes take effect.

When using deployment groups, ensure that you modify the data access service definitions of each slave to reflect database settings for the slave's environment. While the data access service names for all members of a deployment group should match, the other settings usually must be individually tailored to reflect each Real-Time Server host.

See [Deployment Groups](#), on page 22.

Chapter 11: BIO Access

The BIO Access service provides data access to other BIO data on the Infor architecture for use in campaign targeting. The service is created in the Real-Time Server using Real-Time Administrator. Profiles of the BIO type in RT Studio require an existing BIO access service. Once a BIO profile is defined, its attributes can be used in defining offer-targeting criteria the same way as other profile attributes.

See RT Studio online help for additional information about BIO profiles.

The BIO Access service requires both Infor S&S SSO and the Java Adapter. The Single Sign-On (SSO) Server is available through Sales and Service-another Infor Enterprise Marketing Suite application. For the latest information about the version of Sales and Service which is supported with Interaction Advisor, see the Platform Support Matrix document (Platform Support Matrix - Infor Interaction Advisor.doc) available on the Infor Customer Support web site: <http://www.inforxtreme.com>.

Note: You should use the Java Virtual Machine bundled with the J2EE application server which the S&S SSO Server is utilizing, for either WebLogic or WebSphere.

BIO Service Settings

Note: Do not copy and paste scripts, commands, or code from this document. Line breaks and some other characters picked up when you copy from the PDF can cause errors in the pasted text.

Settings for BIO Access services are described in the following paragraphs.

- **Name.** The name used when defining BIO profiles. The name must be unique to the Real-Time Server.
- **Specify Properties option.** When you select this option, you must enter the following properties.
 - **Provider URL.** The JNDI server URL provided by the application server to connect to the JNDI service.
For WebLogic, the URL is `t3://hostname:port`, where `hostname` is the DNS of the machine running the application server, and `port` is the listen port for the server. The default listen port is 7001.
For WebSphere, the URL is `corbaloc:iiop:hostname:port`, where `hostname` is the DNS of the machine running the application server, and `port` is the ejb port for the server. The default ejb port is 2809.
 - **Initial Context.** The JNDI naming factory.
For WebLogic, this is `weblogic.jndi.WLInitialContextFactory`
For WebSphere, this is `com.ibm.websphere.naming.WsnInitialContextFactory`

- **Use Properties File option.** When you select this option, you indicate a properties file that specifies the configuration for the JNDI server.
- **WebSphere EJB Path.** Required only for WebSphere. Leave this field blank for WebLogic.
- **Description.** An optional description of the BIO Access service.

To enable BIO Access services:

- 1 Enable the Java Adapter (See [Java Adapter](#), on page 54), and update the Classpath as outlined below.
 - a Make sure the **Classpath** includes this file from your Interaction Advisor installation:
 - `javaobj.jar` - located in `<Interaction Advisor installation directory>/java/lib`
 - b Make sure the **Classpath** includes paths to files from the Infor Sales and Service installation. If Sales and Service is NOT installed on the same machine as Real-Time Server, first copy these files from Sales and Service to the RT Server machine. Files include:
 - `shared.jar` - located in `<Sales and Service installation directory>/shared/lib` directory.
 - `shared_ejb.jar` located in `<Infor Sales and Service installation directory>/deploy/ejbs/<appserver>` -where `<appserver>` represents the appropriate WebSphere or WebLogic application the S&S SSO server is utilizing.
 - c Make sure the Classpath includes paths to files from the application server installation. Again, first copy these files to the RT Server machine, if not present already:
 - For WebSphere:
 - For WAS 6.1


```
com.ibm.ws.admin.client_6.1.0.jar
com.ibm.ws.webservices.thinclient_6.1.0.jar
```
 - For WAS 7.0


```
com.ibm.ws.ejb.thinclient_7.0.0.jar
com.ibm.ws.orb_7.0.0.jar
```
 - For WAS 8.0


```
com.ibm.ws.orb_8.5.0.jar
com.ibm.ws.ejb.thinclient_8.5.0.jar
```

All of these files are located in: `<Websphere installation directory>/AppServer/runtimes`
 - For Weblogic:
 - `wlfullclient.jar`
- 2 Enable S&S Single Sign-On (see [Server Settings](#), on page 17 and the appropriate guide available for the Single Sign-On Server for more information. The Single Sign-On Server is available through Sales and Service - another Infor Enterprise Marketing Suite application. For the latest information about the version of Sales and Service which is supported with Interaction Advisor, see the Platform Support Matrix document available on the Infor Customer Support web site: Specify this information: <http://www.inforxtreme.com>.
- 3 On the **Settings** sidebar menu, click **BIO Access**.
- 4 On the Add a BIO Access settings page, click **Edit**, then click **add**.
- 5 In the **Name** field, enter a name for the service that is unique in Interaction Advisor.

- 6 If you have a Java properties file for your JNDI server:
 - a Select the **Use Properties File** option.
 - b In the **Properties File**, enter the full path and file name for your file.
- 7 If you do not have a properties file, select the **Specify Properties** option and enter the **Provider URL** and **Initial Context** for your JNDI server. These fields are case sensitive.
- 8 If you are using WebSphere, enter the following for the **WebSphere EJB Path**, replacing nodename and servername with the appropriate values for your system:


```
cell/nodes/nodename/servers/servername/.
```

 These fields are case sensitive.
- 9 Optional: Enter a **Description** for the BIO Access service.
- 10 Click **OK**.
- 11 Restart the Real-Time Server.

BIO Access Error Messages

The following table, where JA is a short for Java Adapter, lists the error messages displayed when BIO access is not properly configured at the startup of Real-Time Server and during BIO access.

If	Can use JA?	Can use BIO ?	Can create BIO service or engine using Real-Time Administrator?	Error message displayed at startup of Real-Time Server?	BIO access error message
JA is disabled	No	No	Yes	None	Java adapter is disabled.
JavaObj.jar is missing	No	No	Yes	Cannot find helper classes. Include path to JavaObj.jar in the environment variable CLASSPATH.	Java adapter is disabled.
shared.jar is missing (JVM is set in JA, and JavaObj.jar is in CLASSPATH)	Yes	No	Yes	None	Cannot load BIO helper classes because shared.jar is not specified in CLASSPATH.

If	Can use JA?	Can use BIO ?	Can create BIO service or engine using Real-Time Administrator?	Error message displayed at startup of Real-Time Server?	BIO access error message
JVM cannot be loaded	No	No	Yes	Cannot load JVM.	Java adapter is disabled. Attempting to do BIO fetch will cause an BIO error.

Chapter 12: Java Adapter

The Real-Time Server's Java Virtual Machine (VM) must be enabled and configured if campaign packages the Real-Time Server handles will include JavaScript functions that contain Java classes.

You must use a Java Virtual Machine. For the latest information about which versions of the JVM/JRE are supported, see the Platform Support Matrix document available on the Infor Customer Support web site: <http://www.inforxtreme.com>.

Java Adapter Settings

Java Adapter settings and their defaults are detailed as follows:

- **Enable Java Adapter.** Yes indicates that the VM for the Real-Time Server is enabled. When enabled, the VM starts and stops with Real-Time Server. Real-Time Server uses the Java Native Interface (JNI) to communicate with the VM.
- **Virtual Machine library.** The full path to the Java VM DLL or shared library, for example:
 - Windows: <JAVA_HOME>\jre\bin\server\jvm.dll
 - Linux: <JAVA_HOME>/jre/lib/amd64/server/libjvm.so
- **Classpath.** The classpath setting for your VM. Refer to your VM documentation VM for details. Ensure that the classpath points to the jar file named `javaobj.jar`, which contains the helper class `com.epiphany.rpjavaobj.Helper`. You need not specify the path to the Java core classes. To add the files that are on a different machine to Classpath, map a drive to the remote machine (Windows) or mount it (Linux).

When listing multiple `.jar` files for the ClassPath, separate each `.jar` file as follows:

platform:	separator:
Windows	; (semicolon)
Linux	: (colon)

- **Windows example:** `C:\Program Files\Epiphany\Java\lib\javaobj.jar; C:\Program Files\Epiphany\Java\lib\myjar.jar`
- **Linux example:** `/opt/Epiphany/RT/Java/lib/javaobj.jar:/opt/Epiphany/RT/Java/lib/myjar.jar`

- **Options.** The VM command line options. Any command line option in the VM documentation is acceptable. However, do not use `-Djava.class.path=path`. Instead, use the classpath setting.

To change Java Adapter settings:

- 1 Click **Java Adapter** in the Settings sidebar menu.
- 2 On the **Java Adapter settings** page, click **Edit**.
- 3 Enter the changes you want, then click **OK**. Your changes take effect the next time the current Real-Time Server starts.

Environment Variables

The following table outlines the environment variables and the paths each needs to contain, if you wish to enable the Java Adapter:

Operating System	Environment Variable	Required Paths	Example
Linux	LD_LIBRARY_PATH	Library Directory	<Java/JDK install dir>/jre/lib/amd64/server
		Parent Directory	<Java/JDK install dir>/jre/lib/amd64
		Native Threads Directory	<Java/JDK install dir>/jre/lib/amd64/server/native_threads

Chapter 13: Real-Time Miners

A real-time miner is a self-learning and predictive algorithm capable of providing offer likelihood predictions in addition to selecting the best offer for a given customer. Real-time miners incorporate the Multidimensional Analytical Profiler (MDAP), based on Bayesian statistics, to maintain customer profiles for individual offers as campaign targeting progresses.

When Real-Time Servers are part of a deployment group, their associated real-time miners can be configured for shared learning across master or slave servers. Distributed real-time miner learning enables a newly deployed slave real-time miner to gain the learning of all other real-time miners in the deployment group upon startup.

See [How Real-Time Miners Work in a Deployment Group](#) on page 60.

This chapter explains how to set up the Real-Time Miner Services which serve the Real-Time Miners in the campaign packages deployed to the Real-Time Server you are working with.

Accessing Real-Time Miner Settings

To access Real-Time Miner settings:

- 1 Click **Real-Time Miners** in the Settings sidebar menu. A list of real-time miner services appears for campaign packages on the Real-Time Server on which you are working.
- 2 Click **Edit** at the bottom of the page to enter edit mode.
- 3 Click **[modify]** to access settings for the real-time miner in which you are interested.

Real-Time Miner Settings

These are Real-time miners settings and their defaults:

Service name

The logical name for a real-time miner service. This name must be unique to the current Real-Time Server. Case does not matter.

Note: Service names must be defined in Real-Time Administrator before real-time miners can be created in RT Studio.

Each real-time miner has its own real-time miner service. Different campaign packages deployed to the same Real-Time Server can use the same real-time miner service if they use the same service name and the same real-time miner definition. During campaign targeting, the real-time miner's information is updated when campaigns in any package that refers to its service name are executed.

Synchronize at startup

When this option is selected, real-time miners are synchronized when the server is started, regardless of other synchronization settings.

Periodic synchronization - Interval

How often, in minutes, you want real-time miner information saved to disk or to the RTDB during campaign targeting. The default is 60 minutes. When the interval is set to 0, there will be no periodic synchronizations.

Information is saved only if it has changed since the last time it was saved. Information is also saved whenever Real-Time Server stops. This setting also defines the frequency of data exchange between real-time miner services running on different Real-Time Servers in a deployment group.

- Specify first synchronization time.
- Start Sync Time: Sets the time for scheduling periodic real-time miner synchronizations, regardless of when the server is restarted or when its settings are modified in Real-Time Administrator. Select hour, minutes, and AM or PM from drop-down lists. The default (the check box is not selected) is 3:00 AM. This time is server time; that is, the time is not converted to GMT.

Significance Threshold

Minimum number of extensions that must occur within a single dynamic bin for that bin to be considered statistically significant. The default is 50, meaning that an input value fitting a bin with fewer than that number of extensions would return an impact of 0 or near 0. In other words, it is not predictive.

Convergence Threshold

Minimum number of times an associated offer must be accepted (within any single acceptance level) before the real-time miner predicts likelihood of acceptance for that offer at that level. The default is 20.

Max. Number of Distinct Values

Maximum number of distinct string values to store for a real-time miner field. The default is 2000. After 2000 different values have been encountered for a field, any value that is different from all 2000 is included as **others**.

Description

A description of the real-time miner service.

Collected Data

A pull-down list displays Offers, Offer Acceptance Levels, or Fields in the real-time miner. You can rename or delete items you select in the resulting lists.

Note: In a deployment group, real-time miner settings, except Start Sync Time, can be modified only through the master server. Changes are pushed to the slave servers in the deployment group.

The Start Sync Time value set on the master server is not pushed to the slave servers. A slave server's Start Sync Time must be changed in Real-Time Administrator on the slave server. This means servers in a deployment group can synchronize their real-time miners at different times.

Creating Real-Time Miner Services

To create a real-time miner service:

- 1 Click **real-time miners** in the Settings sidebar menu.
- 2 On the real-time miners settings page, click **Edit**, then click **add**.
- 3 In the **Service Name** field, enter a logical name unique to the current Real-Time Server.
- 4 If you want the real-time miner to synchronize when the server is started, select **Synchronize at startup**.
- 5 If you want real-time miner information to be saved periodically during campaign targeting, select **Periodic synchronization**.
 - Specify the synchronization interval in minutes.
 - If you want synchronization to begin at a specific time, select **Specify first synchronization time** and enter the time you want synchronization to start.
- 6 In the **Significance Threshold** field, enter the number of times an associated offer must be accepted before the real-time miner predicts likelihood of its acceptance.
- 7 In the **Max. Number of Distinct Values** field, specify the maximum number of distinct string values to store for a real-time miner field.
- 8 (Optional) In the **Description** field, enter a description for the real-time miner service.
- 9 Click **OK** to add the service. The real-time miner service is immediately available for use in campaign packages.

Modifying Real-Time Miner Services

When you access the Modify Real-time miners settings page, all real-time miners settings appear. You can modify any Real-Time service setting except for the service name and maximum number of distinct values.

In addition to these settings, information about the real-time miner associated with the service appears if:

- The real-time miner has been defined in a campaign package deployed to the Real-Time Server you are working with.
- The Real-Time Server you are working with is not a slave in a deployment group.
- At least one offer associated with the real-time miner has been extended during campaign targeting.

Each field in the real-time miner is listed, as well as each associated offer that has been extended at least once.

Most modifications you make to Real-Time services support changes you have already made to the underlying real-time miner and offer definitions using RT Studio.

Note: There are limits to the number of offers that RT can handle. IA Manager takes a long time to refine or make changes to campaign packages when there are more than 1750 offers.

To modify a real-time miner service:

- 1 Click **real-time miners** in the Settings sidebar menu.
- 2 On the real-time miners settings page, click **Edit**.
- 3 Click **modify** in the same row as the real-time miner service you want to edit.
- 4 In the **Collected Data** drop-down list, select the type of data (**Offers**, **Offer Acceptance Levels**, or **Fields** previously defined in RT Studio) for which you want to modify settings.
- 5 If you want to exclude information from an offer, offer acceptance level, or field from Interaction Advisor Manager real-time miner reports, select the appropriate offer, offer acceptance level, or field and click **delete**.
- 6 If you want to rename an offer, offer acceptance level or field, select the appropriate object and click **rename**. You are prompted for the new name.
Note: The offer, level, or field should also be renamed in the campaign packages that use this real-time miner.
- 7 Make any other modifications you like. (You cannot modify the service name and maximum number of distinct values.)
- 8 Click **OK**.

Deleting Real-Time Miner Services

To delete a real-time miner service:

- 1 Click **real-time miners** in the Settings sidebar menu.
- 2 On the real-time miners settings page, click **Edit**.
- 3 Click **delete** next to the service name you want to remove.
- 4 Click **OK** to confirm the deletion. The real-time miner service name is no longer available.

Storing Real Time Miners in an RTDB

You can also store real-time miners in an RTDB. This practice provides the backup and consistency advantages of a database. It also provides a way to share real-time miner learning among instances of the same campaign package deployed to different Real-Time Servers. When you store real-time miners in an RTDB that is used by Real-Time Servers in a deployment group, the real-time miners learn from campaign targeting performed by all members of the group.

To save a real-time miner in an RTDB, enable storage of real-time miners in an RPDB.

See [Saving Packages, Real-Time Miners, and Audit Trails in the RTDB](#) on page 19 for the procedure.

To make a real-time miner sharable among Real-Time Servers:

- 1 Connect with the Real-Time Server you want to designate as master.
- 2 Set up a deployment group. To do so, follow the procedures in [Defining a Deployment Group](#) on page 25.
- 3 On the Server **Settings** page for the master, select **Store RT Miner Models in Database**.
- 4 Any campaign packages currently deployed to the Real-Time Server are copied to the RTDB within seconds. Any real-time miners associated with the campaign package are also automatically stored in the RTDB. Audit trail records are written as soon as any change is made to a campaign package deployed to the current Real-Time Server.

Storing and Sharing Older Real-Time Miners

To store older real-time miners in an RTDB and share them among Real-Time Servers, configure any affected Real-Time Server to enable RTDB storage and set up a deployment group. The existing real-time miners automatically migrate to the RTDB and become enabled for sharing when the Real-Time Server is so configured.

See [Deployment Groups](#) on page 22.

How Real-Time Miners Work in a Deployment Group

The servers in a deployment group can optionally share the knowledge accumulated by their real-time miners through the deployment database. To enable shared learning, select the option **Store RT Miner Models in Database** (see the procedure, “To make a real-time miner sharable among Real-Time Servers,” above). Interaction Advisor supports sharing knowledge only through the database; that is, there is no direct exchange of real-time miner contents between participants of a deployment group. As a consequence, if **Store RT Miner Models in Database** is not enabled, the real-time miners on each server learn at their own pace, and knowledge collected by a real-time miner on any given member of the deployment group is not accessible to the other members.

However, even though it is possible to run a deployment group without sharing the real-time miner knowledge, Interaction Advisor still enforces the consistency of real-time miner definitions across the group, just as it does for other types of services: campaign packages, recommendation models, and the like. To ensure that all real-time miner definitions are consistent, only the real-time miner definition on the master server can be changed. The master notifies the slaves of changes to its real-time miner definitions. Creating a new real-time miner on the master also results in creation of a real-time miner with the same name on each slave. This is necessary because the packages on the slaves are also identical to the packages on the master and refer to real-time miners by name.

This consistency enforcement has an important side effect. If a slave server is taken out of the deployment group (by declaring it stand alone) and then put back in the deployment group, the master pushes its Real-Time Miner definitions onto the slave. This has the consequences described in the

following paragraphs, depending on whether **Store RT Miner Models in Database** is enabled. For simplicity the following discussion assumes that only one real-time miner service is defined on the master (and shared within the deployment group).

With Store RT Miner Models in Database Enabled

While the former slave is running alone, it accumulates knowledge in the real-time miner on top of what it knew before being taken out. When it becomes a slave again, the real-time miner is deleted, and a new real-time miner with the same name is created and populated from the snapshot of the master state stored in the deployment database. It therefore loses all the knowledge accumulated while being free. It is, however, brought up to the current level of knowledge in the deployment group.

With Store RT Miner Models in Database Disabled

During its standalone existence, the former slave continues to learn on top of its knowledge accumulated at the moment of leaving the deployment group. When it is put back in the deployment group, it deletes its RTM service and creates a new one with the name identical to the one defined on the master server. However, since the slave cannot obtain knowledge from the master via the database, it starts learning from scratch.

See [Deployment Groups](#) on page 22 for more information about defining and redefining deployment groups.

Chapter 14: Recommendation Models

The recommendation service within Real-Time Server manages recommendation models built and updated in the RTDB and kept in memory caches for faster access. When a recommender service is configured as model builder, one or more threads update the recommendation model as changes occur and distribute model updates. The incremental model reader, a recommendation service thread, keeps an up-to-date copy of a recommendation model in memory for quick access during campaign targeting.

You access settings for recommendation services by clicking **Recommendation Models** in the Settings sidebar menu. Information about existing recommendation model services used in campaign packages deployed to the Real-Time Server you are working with appears in the Recommendation Models settings page.

Recommendation models and their use during campaign targeting are set up in RT Studio.

Recommendation Model Settings

Recommendation model service settings and their defaults:

- **Model Name:** A recommendation service name.
- **Database Service:** The logical name for the data access service defined for the RTDB used to manage the recommendation model. See the Infor Real-Time Implementation Guide, Part II, for more information about RTDB elements.

This name must be defined in Real-Time Administrator before recommenders that will use this service can be defined in RT Studio.

Model Breadth: An integer parameter that controls the size of the model. Also affects the maximum number of items you can get back as a recommendation for a single input item. The number of recommendations returned is also affected by the number and type of constraints in use.

Infor recommends that you create your recommendation model using the default setting, 50. This setting is low enough to avoid affecting performance and high enough to return 5 to 10 recommendations unless you are using very restrictive constraints. If using the default does not return as many recommendations as you want, try doubling the model breadth number. If doing so does not increase the number of returned products, the model breadth setting is not the reason. In this case, return the number to the default setting.

- **Surprise Factor:** A number between 0 and 1. The higher the surprise factor, the more personalized the recommendations will be. When the surprise factor equals 1, the recommendation model completely discounts the popularity of the output items when deciding what to recommend. In this case, recommendations are completely driven by the affinities between items. Smaller surprise

factor values take the popularity of output items into account. The lower the surprise factor, the more favorite treatment the most popular items receive.

- **Context Weight:** The weight, expressed as a decimal, can be assigned to the interest profile that is an input to the recommendation model as opposed to the demographic profile.
- **Model Builder:**
 - **Server:** The Real-Time Server that keeps the recommendation model current. An asterisk (*) refers to the master server of the deployment group or to the current Real-Time Server when it is not a member of a deployment group. If the current Real-Time Server belongs to a deployment group, the model builder is one Real-Time Server in the group that has been designated to update the recommendation model as changes occur and to distribute model updates to other members of the deployment group.
 - **Port:** The port number of the Real-Time Server that acts as the model builder. The default is 7200.
 - **Number of Threads:** The number of processing threads to be used for recommendation model updates. By default, there is one thread. You can increase this number to support a greater number of concurrent database queries by the model builder.
 - **Minimum Percent Change to Process:** Controls the number of records that must change before the recommendation model is updated. When the percentage of records changed equals or exceeds the percentage you specify, the model is updated. If the minimum has not been reached, the model builder goes back to sleep. This condition is applied to each field. By default, 5 percent.
 - **Maximum Sleep/Work Ratio:** This setting controls how long model builder should wait before starting a new model update if the changes in the input data are below the Minimum Percent Change to Process. The waiting time is determined relative to the amount of time the last model update took. If the last model update took 10 minutes and the Maximum Sleep/Work Ratio is set to 20, the model builder will wait 200 minutes before starting to process input data changes that are below the 5 percent threshold.
 - **Description:** A description of the service.

To create a recommendation model service:

- 1 On the Recommendation Models settings page, click **Edit**.
- 2 Click **add**.
- 3 In the **Model Name** field, enter a recommendation service name.
- 4 From the **Database Service** pull-down list, select a data access service describing an RTDB connection.
- 5 In the **Surprise Factor** field, enter a number between 0 and 1.
- 6 In the **Server** field, enter the name or IP address of the model builder. If it is not a member of a deployment group, enter an asterisk (*). In the **Port** field, enter the port number used by the Real-Time Server that acts as the model builder.
- 7 In the **Number of Threads** field, define the number of processing threads to be used for recommendation model updates.
- 8 In the **Minimum Percent Change to Process** field, specify the percentage of records that must change before the recommendation model is updated.
- 9 In the **Maximum Sleep/Work Ratio** field, specify the maximum sleep/work ratio.

Chapter 15: Campaign Packages

You access Campaign Packages settings by clicking **Campaign Packages** in the Settings sidebar menu. Current settings for each campaign package deployed to the current Real-Time Server appear on the Campaign Packages settings page.

Campaign packages can be deployed using only Real-Time Studio or Interaction Advisor Manager. Refer to the online help systems for these products for more information.

Campaign packages can be deployed as a test version or a production version. A test version is a campaign package intended for limited use, by certain Interaction Advisor client application users or by Interaction Advisor Manager users. A production version is a campaign package that is available for widespread use by Interaction Advisor client applications. Either of these versions can be locked for refinement using Real-Time Studio or Interaction Advisor Manager. When a package is refined, it is locked using the name of the user making refinements and a copy is made to use for changes. A locked campaign package cannot be deleted, but you can transfer a campaign package lock to another user.

The service names of test version campaign packages end in the suffix `_Test`. When the test version is deployed from Interaction Advisor Manager, the user name is added to the suffix. For example, if the service name of a production version campaign package is `MyPackage`, when John Doe deploys it for testing from Interaction Advisor Manager, its corresponding test version service name is `MyPackage_Test_jdoe`. The suffix `_Edit` is added to package copies that are being refined.

Loading Packages at Deployment

Campaign packages can take awhile to load if, for example, custom code must initialize cached profiles from the database. In such cases, the first events since a package was deployed may time out because a package load takes longer than the maximum time an event is allowed to take. To ensure that events do not time out, you can specify that production campaign packages are to be loaded at the time they are deployed.

You can set a minimum number of campaign package instances to be loaded when a new version of a production package is loaded. Two settings, one for master and another for standalone servers, specify the minimum number of instances for all production campaign packages.

- **Minimum Campaign Packages Instances.** Specifies the minimum number of campaign package instances to be loaded on a master or standalone server when a new version of a production package is loaded.

On master servers, it is recommended that the minimum number of instances be set to zero.

- **Minimum Campaign Packages Instances on Slaves.** Specifies the minimum number of campaign package instances to be loaded on slaves when a new version of a production package is loaded. This setting appears only if the server is a master.

On slave servers, it is recommended that the minimum number of instances be the same as the number of server socket threads, the main threads processing events, or the same as the number of server socket threads plus the number of command threads (command threads can also process events when there is high load).

Changing Minimum Package Load Settings

- 1 On the Campaign packages settings page, click **modify**.
- 2 Make your changes, then click **OK**.

Campaign Packages Settings

The basic information presented for the Campaign Packages is outlined below:

Service Name

A name identifying the service name and version of a deployed campaign package.

Description

A description of the campaign package specified when it was deployed.

Being Edited By

The name of the user who has the package locked for refinement. More than one user may be working on different parts of the package at a time. Thus, the **Being Edited By** column may contain more than one user name.

Copy to Server

Copies a Campaign Package from one server to another server. This field is used to migrate a campaign package from development to production instances.

Note: This option is displayed only for the users with the administrator's role

User Editing Session

To discard the changes of a user's editing session:

- 1 On the 'Settings' sidebar menu, click **Campaign Packages**.

- 2 Click **Edit** on the Campaign packages settings page, then click **Edit** in the same row as the package from which you want to discard a user's changes.
- 3 Click on **discard changes** in the same row as the user whose changes you wish to discard.

Copy to Server

Use this functionality to copy a campaign package from one server to another. You can copy the campaign package without accessing the Real Time Studio or the command prompt. This option is displayed only for the users with the administrator and package master roles. The current user with administrator and package master role also exists on destination server with those same roles.

To copy a campaign package:

- 1 Select a campaign package.
- 2 Right-click and select **Copy To Server**. The **Copy to Server** page is displayed.
- 3 Specify this information:

Destination Server

The name of the server to which the campaign package is copied.

Port

The port number of the server to which the campaign package is copied.

Secure

If this check box is selected, a secured URL (https://) is used to connect to the destination server.

Replace Existing

This option indicates what the server will do if the package exists on the destination server.

Following are the options:

- **No:** If the package exists on the destination server, the package is not copied.
- **Yes:** The package is replaced on the destination server.
- **Replace and Delete Sessions:** The package is replaced, and existing sessions are stopped.

- 4 Click **Submit** to copy the campaign package to the destination server. To copy the package, you must have access to the destination server.

Campaign Package Deployment Tools

This section describes the getpackage and the deploypackage tools.

Using getpackage tool

The `getpackage` tool, a utility program located in the installation directory (for example, `C:\Program Files\Epiphany\RT\`), can be used to export a campaign package in Real-Time Server. The usage of `getPackage` is as follows:

```
getpackage -? | <user> <pw> <package> <host> [-p <port>] <file>
```

That is, at the command prompt, you can enter either

```
getpackage -?
```

or

```
getpackage <user> <pw> <package> <host> [-p <port>] <file>
```

Here are the meanings of the command-line arguments:

Command-line Argument	Description
-?	Shows usage of <code>getpackage</code>
[]	The argument is optional
user	Username used to login to Real-Time Server
pw	Password used to login to Real-Time Server
package	Name of package in Real-Time Server to be exported
host	Hostname or IP address of the machine where Real-Time Server runs
port	Port number of Real-Time Administrator, default 7200
file	Filename of the exported package. Note: If this value is defined to be within the 'Program Files' directory structure, the command prompt will need to be run as Administrator.

For example, the following exports the campaign package `Golden` in Real-Time Server running on host **sunshine** to file `C:\packages\gf.mop`:

```
getpackage joe joepass Golden sunshine C:\packages\gf.mop
```

Using deploypackage tool

To deploy a campaign package to a specific Real-Time Server, use the `deploypackage` tool, a utility program located in the installation directory. The usage of `deploypackage` is as follows:

```
deploypackage -? | <user> <pw> <package> <host> [-p <port>] <SOURCE>
```

That is, at the command prompt, you can enter either

```
deploypackage -?
```

or

```
deploypackage <user> <pw> <package> <host> [-p <port>] <SOURCE>
```

where:

- SOURCE: <FILE> | <SERVER>
- FILE: -f <file>
- SERVER: -s <sPackage> <sHost> [-sp <sPort>] [<SOURCE_USER>]
- SOURCE_USER -sUser <sUser> <sPasswrod>

Here are the meanings of the command-line arguments:

Command-line Argument	Description
-?	Shows usage of <code>deploypackage</code>
[]	The argument is optional
user	Username used to login to the destination Real-Time Server
pw	Password used to login to the destination Real-Time Server
package	Name of package to be deployed to the destination Real-Time Server
host	Hostname or IP address of the machine where the destination Real-Time Server runs
port	Port number of destination Real-Time Administrator, default 7200
-f	Indicates that path of source file follows, getting package contents from a specified source file
file	Path of source file containing the package to be deployed
-s	Indicates that information about the source Real-Time Server follows, getting package contents from a specified source Real-Time Server
sPackage	Package name on the source Real-Time Server
sHost	Hostname or IP address of the machine where the source Real-Time Server runs
sPort	Port number of source Real-Time Administrator, default 7200
-sUser	Uses username and password, different from those used to login to the destination Real-Time Server, to login to the source Real-Time Server
sUser	Username used to login to the source Real-Time Server

Command-line Argument	Description
sPassword	Password used to login to the source Real-Time Server

Chapter 16: Package Archiving

Campaign packages can be archived automatically or manually. A campaign package archive records the package's state, which includes the package's fields and campaigns as well as offers in the campaigns.

Auto Package Archiving

To enable auto package archiving:

- 1 Start Real-Time Administrator.
- 2 Select **Campaign Packages** in the left navigation pane.
- 3 Click **modify....**
- 4 In the screen for editing campaign package settings, select the **Archive Enabled** check box, specify a folder (a directory name) after the **Archive Location** field, and select one or both the archive formats.

See [Auto Package Archiving](#) on page 70.

- 5 Click **OK**.

The changes to the auto package archiving settings are made by the Real-Time Server immediately. If you use Real-Time Studio to deploy a package, you can clear the **Archive** check box to disable auto package archiving. RT Server considers the **Archive** checkbox setting at the package level that is last used in RT Studio, and this setting is applied when packages are deployed using IA Manager.

A campaign package is deployed to production on either a master or standalone Real-Time server. The campaign package is saved in the archive location specified in **Campaign Package Settings** page on the **Real-Time Administrator** application. The format of archive files can be either, an XML or MOP file. The naming convention of archive files depends on the selected archive format in Campaign Package Settings page on the Real-Time Administrator application. Example: Either `PackageName_Version_deployerUsername.xml` or `PackageName_Version_deployerUsername.mop`.

Note: The file name include the log in ID of the user who archives the package.

An XML-formatted archive contains only a subset of the campaign package deployed to production. The subset includes package fields, campaigns, and offers and their functions. If you need more information on the state of the package, you must choose the MOP archive format.

Manual Package Archiving

Using Real-Time Studio, you can manually archive a package by exporting it to a MOP or XML file. Package deployment is not required for manual package archiving, which can be done even if Real-Time Administrator disables auto package archiving.

Chapter 17: Logging

The log files `moperror.log` and `system_trace.log` are created automatically the first time you start Real-Time Server. The files are tab-delimited ASCII files located in one of the following:

- Windows: `C:\Program Files\Epiphany\RT\LogFiles`
- Linux: `<InstallDir> /LogFiles`

You can create or modify ASCII log files for Real-Time Server to use. Information about existing logs appears on the **Logging settings** page.

A Linux-specific performance monitoring utility is also available. You can run the utility in interactive mode or in the background to create a log file.

See [Performance Monitoring on Linux](#) on page 80

Note: Do not copy and paste scripts, commands, or code from this document. Line breaks and some other characters picked up when you copy from the PDF can cause errors in the pasted text.

The CAMPAIGNPACKAGE ERROR Log

The CAMPAIGNPACKAGE ERROR log event is associated with the ASCII file `moperror.log`. Records are written to this log file automatically by Real-Time Server when a problem exists in campaign packages or Real-Time Server command files, for example:

- A lookup failed, perhaps because a database table could not be accessed or a field name used in targeting criteria does not exist.
- The current event does not have associated with it an expected event field.
- A JavaScript error occurred.

You can also write your own log records to this log. The columns in the following table are predefined for CAMPAIGNPACKAGE ERROR:

Column	Description
ERRORMSG	A message describing the failure that occurred
ERRORLOCATION	The line number on which the error was detected
CAMPAIGN_PACKAGE	The service name of the campaign package that failed

The SYSTEM_TRACE Log

By default, SYSTEM_TRACE is turned off. To turn it on, you must contact Infor Support.

SYSTEM_TRACE corresponds to the ASCII file `system_trace.log`. SYSTEM_TRACE adds tracing messages for data access, the MDAP engine, and possibly other engines. Data access trace contains SQL queries submitted to the database server. For the MDAP engine, these trace messages document the real-time miner's key operations, such as offer /field updates and deletions and miner synchronizations. In addition to EVENT_TIME and EVENT_TIME_SEC., SYSTEM_TRACE also includes a set of common columns, so that different engines can easily add traces to the same file.

Note: The exact column names shown must be used. Do not modify them.

A trace message need not have information in all columns.

Column	Description
START_TIME_MS	Start time of the trace call in milliseconds since 1/1/70.
DURATION_MS	Duration of the trace call in milliseconds.
THREAD_ID	Thread ID on which the trace call is executed.
DATA_ACCESS_THREAD_ID	Data access thread ID on which the trace call is executed.
SESSION_ID	Session ID for the event.
ENGINE	DA, MDAP, RE, TE, JS, and the like.
SERVICE	Service name of the engine.
CLASS_ADDRESS	The address of the class originating the trace call.
CATEGORY	The tracing category; an engine-specific category for filtering.
COMMAND_INPUT	Input parameters to the command.
COMMAND_OUTPUT	Output result of the command.
MSG	Additional information that does not fit in the other columns.

Logging Settings

You can access logging settings by clicking **Logging** in the **Settings** sidebar menu. Logging settings and their defaults are:

Log Event Name

The logical name associated with a log file for a particular event. Log event names are case sensitive and must be unique to a Real-Time Server.

Log File Name

The ASCII file to be associated with the log event. Enter the full path name as well as the file name. If it does not have an extension, the extension `.log` is used.

Append to Log File

Yes indicates that log file contents are preserved at Real-Time Server startup; new records are appended to the end of the file. **No** indicates the log file is recreated at startup.

Column Separator

Data is written to the log file in the format of multiple columns separated either by tabs or commas.

Changing a separator and restarting Real-Time Server deletes all information from the log file.

Creating a Log Record

You can create a log record using the `LogEvent()` function in campaign package JavaScript functions.

See the Interaction Advisor JavaScript Reference online help system for more information.

The target log event and the information to log to particular log file columns are defined in arguments passed in the function, for example:

```
LogEvent("SYSTEM_TRACE", "ERRORMSG=" + "Age:" + retval);
```

The two fields in the following table are written by Real-Time Server to each log record it writes; these fields are not visible in Real-Time Administrator, but appear in the log files themselves as the first two columns:

Field	Description
EVENT_TIME	The time the error was logged, for example, Tue May 18 15:17:37 1999.
EVENT_TIME_SEC	The number of seconds since host startup the error was logged, for example, 927065857.

Defining Custom Logs

You can also define custom logs. To define a log, you must first add a log event.

To add a log event:

- 1 Click **Logging** in the **Settings** sidebar menu.
- 2 On the **Logging settings** page, click **add**.
- 3 Specify these details:

Log Event Name

The name of the event.

Append to existing log

Select this check box to preserve the log file contents when Real-Time Server starts. Otherwise, the log is deleted and a new one created each time Real-Time Server starts.

Server File Name

The path and ASCII file name you want to associate with the log event.

Separator

The character to delimit log record values.

- 4 Click **OK**.
- 5 To define a column for your file, click **add**.
- 6 On the **Logging settings - Add a column** page, specify the **Column Name**, the **Column Type**, and the **Column Length**.
- 7 Click **OK** to save the column definition.
- 8 Repeat steps 7 through 9 for each column you want in the log file.
- 9 When you are finished defining columns, click **Done**. The log file is created the next time Real-Time Server starts.

Call `LogEvent()` from a campaign package function to assign values to log record columns and write the record to your log:

```
LogEvent("MyLogEvent", "Column1 = col1Value", "Column2=col2Value");
```

Before targeting the campaign within which `LogEvent()` executes, restart Real-Time Server to register the log event.

Modifying Logs

Changes you make to log settings take effect the next time the current Real-Time Server restarts.

To delete a log event:

- 1 Click **Logging** in the Settings sidebar menu.
- 2 On the Logging settings page, click **Edit**.
- 3 In the same row as the log event you want to modify, click **delete**.

Modifying a log event

To modify a log event:

- 1 Click **Logging** in the Settings sidebar menu.
- 2 On the Logging settings page, click **Edit**.
- 3 In the same row as the log event you want to change, click **modify**.
- 4 Change the settings, then click **OK**.

To delete a log event column, click **delete** in the same row as the column name you want to delete, then click **OK**.

Chapter 18: Performance Monitoring Tools

This chapter explains how to monitor Real-Time Server performance in Windows and Linux environments.

Performance Monitoring on Windows

You can use Windows Performance Monitor tool to display or log performance counters characterizing various aspects of Real-Time Server performance. To access Performance Monitor from the Windows Start menu, select **Start > Programs > Epiphany > Interaction Advisor > Performance Monitor**. By default, Real-Time Server's performance counters are available after the installation of Real-Time Server.

For Windows 2012, select **Start > Apps > Performance Monitor**.

If the counters do not appear in the Performance Monitor console:

- 1 Unregister any existing Real-Time Server counter by executing the program `UnRegisterDmPerf.bat` located in the `config` subdirectory of Real-Time Server's installation directory.
- 2 Re-register the counters by executing the program `RegisterDmPerf.bat` located in the same `config` subdirectory.

Note: If a log using the counters will not start, verify appropriate Logon Settings for the logging service. Open **Start > Programs > Administrative Tools > Services**. Open Properties for the Performance Logs and Alerts service and verify the Log On setting has Local System account selected.

Counters

All available counters are described in the following sections.

Performance Object: Real-Time Server General

This set of counters provides detailed information on Real-Time Server's queues as well as on general performance. It contains the most useful counters for performance monitoring and server parameter

tuning. Socket requests are events to be processed (sent by clients), SOAP requests, requests from Interaction Advisor Manager and Real-Time Administrator, and a few other administration commands (inter-deployment group communication, for example).

- Internal Queue Length: Current number of commands waiting to be served by the internal (command) threads.
- Internal Queue Peak: Maximum number of commands waiting to be served by the internal (command) threads ever reached since Real-Time Server was started.
- Internal Threads Active: Current number of internal (command) threads actively processing commands.
- Internal Threads Peak: Maximum number of internal (command) threads actively processing commands ever reached since Real-Time Server was started.
- Services Active: The number of packages or other deployable services currently in memory.
- Sessions Active: Current number of active TargetingEngine sessions (started, no wrap-up yet).
- Sessions Peak: Maximum number of simultaneously active TargetingEngine sessions ever reached since Real-Time Server was started.
- Socket Queue Length: Current number of socket requests waiting for an available socket thread.
- Socket Queue Peak: Maximum number of socket requests waiting for an available socket thread ever reached since Real-Time Server was started.
- Socket Requests / Sec: Number of socket requests arriving each second.
- Socket Requests Total: Total number of socket requests arrived since Real-Time Server was started.
- Socket Response Time: Average time to receive, enqueue, execute and respond to socket requests.
- Socket Threads Active: Current number of socket threads actively processing socket requests.

Performance Object: Real-Time Server Engines

This set of counters provides detailed information on the load of each Real-Time Server engine. Commands include session wrap-ups, background tasks, events to be processed (under load only), and the like.

- Command Time: Current average command (or event) processing time (including wait time) by the selected Real-Time Server engine(s).
- Commands / Sec: Current average number of commands (or events) processed per second by the selected Real-Time Server engine(s).
- Socket Requests / Sec: Current average number of socket requests processed per second by the selected Real-Time Server engine(s).
- Socket Requests Total: Total number of socket requests processed per second by the selected Real-Time Server engine(s) since Real-Time Server was started.
- Socket Response Time: Average time to receive, enqueue, execute and respond to socket requests by the selected Real-Time Server engine(s).

Instances (Real-Time Server Engines)

- AdminEngine: Processes commands.
- DataAccessEngine: Processes commands.
- HTTPEngine: Processes socket requests.
- LogEngine: Processes commands.

- MdapEngine: Processes commands.
- RecommendationEngine: Processes commands.
- ServerEngine: Processes commands.
- TargetingEngine: Processes socket requests and commands.

Performance Object: Real-Time Server Interfaces

These counters provide traffic information for the port interfaces of Real-Time Server:

- Accepts: Total number of socket requests accepted by the selected Real-Time Server interface(s) since Real-Time Server was started.
- Accepts / Sec: Current average number of socket requests accepted per second by the selected Real-Time Server interface(s).
- Overruns: Total number of socket requests for the selected interface(s) that have been discarded because Real-Time Server's socket request queue was overrun since Real-Time Server was started (the size of this queue is set by the Maximum Events Queued property on the Server page of Real-Time Administrator).
- Port Number: Port number for the interface (as defined in the Port Interface Settings section of Real-Time Administrator).
- Timeouts: Total number of socket requests for the selected interface(s) that have exceeded this interface's (Service Guarantee) Timeout (set by the Timeout property on the Port Interface Settings page of Real-Time Administrator) since Real-Time Server was started.

Instances (Port Interfaces)

- Admin
- Admin_HTTP
- Admin_HTTPS
- Event
- Event_HTTP
- Event_HTTPS
- Event_SSL

Performance Object: Real-Time Server Services

Services include packages in memory, data access services, real-time miner services, and the like.

- Commands / Sec: Current average number of events processed per second by the selected package(s) or of commands processed per second by the selected DataAccess service(s).
- Commands Total: Total number of events processed by the selected package(s) or of commands processed by the selected DataAccess service(s) since Real-Time Server was started.
- Response Time: Average time the selected package(s)/DataAccess service(s) take(s) to execute and respond to an event/command.
- Service Ref Counts: Current number of events/commands referencing the selected package(s)/DataAccess service(s)

Instances (Running Services)

- DataAccessEngine Services
- TargetingEngine Services (packages)

Performance Monitoring on Linux

You can use the Linux-specific performance monitoring utility `dmperfux` to display or log performance counters characterizing various aspects of Real-Time Server performance. The version of `dmperfux` utility must match the version of Real-Time Server.

Real-Time Server maintains performance counter information in a shared memory segment. The `dmperfux` utility gets the information from this shared memory segment without synchronizing directly with the Real-Time Server. This decoupled architecture ensures that the utility does not adversely affect Real-Time Server performance.

When you run the `dmperfux` utility in display (non-background) mode, it lists all Real-Time Server instances running on the host machine. To select an instance to monitor, type the number of the instance and press the ENTER key. Display mode requires a terminal window compatible with VT100 terminal. To navigate in the display between the menu and the main display area, press **TAB**.

To run `dmperfux`, refer to the following table for command option information or type `dmperfux -h` to display command line options.

You can use the `dmperfux -h` command to display command line options. The commands to run `dmperfux` are:

Command	Description
-h	Display this usage description.
-b	Background mode. Disables all interactions with the terminal window. Used with -l option. The command must terminate with an & (ampersand).
-e	Changes the time stamp in the log file to epoch. Unix time (also known as POSIX time or UNIX Epoch time) is a system counting the number of seconds since 00:00:00 (UTC), January 1, 1970.
-i	Specifies the location and name of the registry file for a running Real-Time Server instance. Used to select between multiple Real-Time Server instances running on the same machine.
-l	Specifies a log file. The log file contains tab-separated values and is suitable for importing to MS Excel or a database.

Command	Description
-o	Specifies an object to be shown. The available objects are "Server", "Interfaces", "Engines", and "Services". Multiple -o options can be used to select multiple objects. If no -o option is specified, all four performance objects are shown.
-u	Specifies update interval in seconds. The update interval determines how often the display is updated and how often a new record is added to the log file.
-a	<p>Specifies an AWS configuration file that comprises an output file and one or more performance counters. For each of these performance counters, the namespace, the metric-name, and unit values must be specified without any spaces.</p> <p>Note: A separate mechanism is required to invoke the aws cli and pass each record from the aws.out file to the cli.</p> <p>Note: This command is only available for Linux. See <i>AWS Cloudwatch documentation</i> for details. This is sample content for a configuration file:</p> <pre>[OutputFile] name=/tmp/aws.out [Server: Internal Queue Length] namespace=IAServer metric-name=Command_Queue_Length units=Count [Server: Socket Response Time] namespace=IAServer metric-name=Response_Time units=Milliseconds</pre>

Note: Do not copy and paste the scripts, commands, or code from this document. Line breaks and some other characters picked up when you copy from the PDF can cause errors in the pasted text.

Use this command to invoke the utility:

```
dmperfux [-h] [-b] [-e] [-i registry-file] [-l log-file] [-o object] [-u
update-interval] [-a aws-config-file]
```

Chapter 19: Troubleshooting

Consult this chapter to review the characteristics of the Real-Time Server and clients in use if difficulties arise when campaigns are executed.

Real-Time Server Setup

The following paragraphs explore possible deployment and resource access problems.

Campaign Package Deployment

Ensure that the campaign package containing the campaign has been deployed to the Real-Time Server being accessed and that the campaign package file exists in the Real-Time Server's campaign package deployment directory.

Data Access

Ensure that all database tables or views on which lookup sources, stored procedures, and tables are based are in locations that can be accessed from the Real-Time Server host. Also ensure that database access privileges shown in the following table for the various kinds of campaign package elements are satisfied by the data access service defined for the database:

To support this element	The Data access service user name must have
Lookup source	CONNECT privilege for the target database SELECT privilege for the target table.
Table source	CONNECT privilege for the target database SELECT, UPDATE, DELETE, and INSERT privileges for the target table.
Stored procedure source	CONNECT privilege for the target database EXECUTE privilege for the target stored procedure.

RT Client Application Setup

The following paragraphs explore possible client connectivity and privilege characteristics problems.

Client Connectivity

Ensure that any campaign package service name specified matches the service name for a deployed campaign package. If a campaign package service name is not identified by client applications, ensure that default campaign package identities are set up correctly using Real-Time Administrator. Also ensure that the port the application is using is the port Real-Time Server is configured to listen on for the particular client. If the client uses Web/HTTP or Socket Component interfaces, ensure that the necessary Real-Time Server command files are accessible.

Database Permissions

Any database activities a campaign package uses rely on the permissions of the user specified in the data access service configuration. Check user permissions on the database server to ensure that they are sufficient for particular activities.

Memory Usage

Cached Profiles

Big memory usage is generally caused by huge cached profiles (adjust Profile Age and Cache Size parameters of the cached profiles in Real-Time Studio, to avoid unlimited caching) or too many sessions in memory (make sure the session idle timeout in Real-Time Studio is set to a reasonably low value).

Fast Reset Performance option

For a specific event, if the number of activated campaigns and offers is relatively small, compared to the size of the package, you can enable the Fast Reset option to improve the system-related performance.

See the *Infor Interaction Advisor Implementation Guide* for details.

CommandQueueYellowMark

This feature exists in order to protect the RT Server from failure due to insufficient virtual memory and/or CPU which handles Event load and Session Wrapup activities. Most systems probably have enough memory and therefore don't need to enable this feature, so it is not enabled by default.

See the Real-Time Implementation Guide for more information about how to enable this feature.

Windows Error Reporting

Windows Error Reporting (WER) can be configured so that full user-mode dumps are collected and stored locally if the RT Server application is crashing.

See Microsoft Windows documentation for details on enabling and using this feature.

Chapter 20: Debugging

This chapter explains how to configure settings for the ScriptEase Workshop JavaScript debugger. This easy-to-use debugger enables you to step through JavaScript code as it is being executed by the server, so you can understand and fix errors in campaign execution. You can use the ScriptEase Workshop debugger to debug:

- A campaign package when sending events from RT Studio.
- A campaign package when sending events from a RT client.
- A Real-Time Server command file used by a RT client.
- A campaign package used by a Real-Time Server command file.

ScriptEase Workshop is available from RT Studio as well as from the installation directory in Windows Explorer (`..\Epiphany\RT\workshop.exe`).

For information on using the ScriptEase Workshop debugger, refer to Real-Time Studio online help and ScriptEase Workshop online help. For information on Real-Time Server command files, refer to chapter 12 in the Real-Time Implementation Guide. The command files contain JavaScript commands Real-Time Server uses to process events that arrive from RT clients implemented using the Web/HTTP and Socket Component interfaces.

You access Debugging settings by clicking **Debugging** in the **Settings** sidebar menu.

Debugging Settings

Debugging settings and their defaults are detailed here:

Disable Debugging

Turns off debugging at the server, making debugging unavailable in Real-Time Studio and ScriptEase Workshop.

Enable Local Debugging

Indicates whether the debugger is available for use on demand from the RT Studio installed on a Windows computer having the `Debug Console IP Address`. During debugging, ScriptEase Workshop must be running on the specified system. For local debugging, the address is `127.0.0.1`.

Enable Remote Debugging

- **Remote Debugger IP Address:** The IP address of the machine running the debugger.
- **Debugger Port:** Identifies the port ScriptEase Workshop monitors for debug sessions. By default, set to 57005.

Note: Because debugging degrades performance, debugging should be enabled only when you want to debug code. In production mode, debugging should be disabled.

Changing Debugging Settings

To modify a Debugging setting:

- 1 In the Settings sidebar menu, click **Debugging**.
- 2 On the Debugging settings page, click **Edit**.
- 3 Make your changes in the appropriate fields.
- 4 Click **OK**. Changes you make take effect the next time the Real-Time Server to which you are connected is restarted.

Chapter 21: Admin Variables

This chapter explains how to manage data to be used by RT Server for accessing external web services.

To avoid the necessity of hard-coding confidential and/or sensitive data in JavaScripts, such as user name and password, the System Administrator can create and manage this data in RT Administrator. These values are then accessible to system functions in IA to be incorporated into API messages.

Both plain text and encrypted values can be entered by the user. The System Administrator creates a name to identify each value, can enter an optional description, and can indicate if the value should also be encrypted by IA when it is stored in RT Server. The name assigned by the System Administrator is how the JavaScript `GetAdminVariable()` method in RT Studio retrieves the values.

To View Admin Variables

To access Admin Variables click **Admin Variables** in the Settings sidebar menu. A list of the defined variables appears in the Admin Variables settings (Editing) page.

To Add Admin Variables

To add an Admin Variable, click **add** on the Admin Variables settings page to display the **Add an Admin Variable** page.

Name

Specify text to identify the value. The field must not be empty and cannot contain any spaces, back slash, forward slash, double quotes or non-printable characters, and must be unique to other Admin Variable names.

Encrypted

Select this check box, to have IA encrypt the value when it is stored in RT Server. In addition to being encrypted when stored in RT Server, the value will be displayed (masked) as ***** when viewed in RT Administrator. Five asterisks are displayed regardless of the actual number of characters. The RT Server encryption is only applicable when the value is stored in RT Server. RT Server automatically decrypts the value when it is retrieved to be sent to the web service.

Value

A string of plain text or a string of encrypted characters.

Description

Optional. This information is displayable in RT Studio and may be helpful to the RT Studio user when working with JavaScript coding.

Click **OK** to save the definition and make it available to RT Studio.

To Modify Admin Variables

- 1 On the Admin Variables settings page click **modify** in the row of the value you want to modify. The 'Modify an Admin Variable' page will appear.
- 2 Make your changes, and then click **OK** to save them. Changes will be effective immediately. The name field cannot be modified.

To Delete Admin Variables

- 1 On the Admin Variables settings page click **delete** in the row of the value you want to delete.
- 2 Click **OK** to confirm the deletion.

Glossary

.NET Client

The client you can use to integrate Interaction Advisor campaign support into client applications that can take advantage of .NET Framework.

accepted cost

The offer characteristic describing how much it costs when the offer is accepted.

activation

The process of making an Interaction Advisor campaign available for use by Interaction Advisor client applications. To describe when to activate a campaign, you use activation conditions. See [activation conditions](#).

activation conditions

Criteria determining when an Interaction Advisor campaign is available for use. When its activation conditions are true, a campaign can be used by Interaction Advisor client applications. In a campaign hierarchy, the activation conditions of campaigns above a targeted campaign must also be true.

Activation conditions can be used, for example, to control the season or month during which a campaign is available, or to regulate when campaigns can be targeted based on call center activity level. See [activation function](#) on page 89.

activation function

The definition you specify in Real-Time Studio to set up campaign activation conditions. An activation function returns true or false. When it returns true, a campaign is available for use.

Administrator rights

The process of evaluating arbitration conditions. See [activation conditions](#).

arbitration

The process of evaluating arbitration conditions. See [activation conditions](#).

arbitration conditions

The criteria governing which offer or offers to extend to a customer when the customer is entitled to receive more than one offer. The offers are ranked using a strategy you specify. The user defined number of the highest ranking offers is returned to the Interaction Advisor client application. See [ranking strategies](#).

You can specify two levels of campaign arbitration conditions: those to evaluate after individual campaigns have been executed (see [activation function](#) and [arbitration strategy](#)) and those to evaluate after all campaigns associated with a particular customer event have been executed (see [final arbitration function](#)). In a campaign hierarchy, arbitration conditions associated with each campaign above each targeted campaign are applied, from lowest campaign to highest campaign in the hierarchy.

arbitration function

A definition used in Real-Time Studio to set up campaign level offer arbitration conditions. An arbitration function returns an ordered list of offers to extend after a campaign is executed. Contrast with [arbitration strategy](#) and [final arbitration function](#).

arbitration strategy

An alternative to definition of campaign level conditions that allows for grouping, sorting and prioritizing of offers.

An arbitration strategy can be used instead of an arbitration function or final arbitration function to define rules for grouping and scoring to rank qualified communications and deciding which one to extend to the current customer. Unlike an arbitration function, the arbitration strategy can be edited by the Manager user.

attribute

See [offer attributes](#) and [profile attribute](#).

audit trail

Tables in the Real-Time database that can be used to provide a change history for campaign packages.

authentication

The process of ensuring a client application that a particular Real-Time Server is authentic, performed using an electronic certificate issued to the Real-Time Server by a Certificate Authority. Also the process by which a user's password is authenticated.

automatic offer tracking

The ability of Real-Time Server to automatically record information for offers extended and accepted using a predefined table named OfferTracker or a predefined event named OfferAccepted. See [OfferTracker table](#) and [offer tracking](#).

backup Real-Time Server

The Real-Time Server in a deployment group that is used when another Real-Time Server in the group is not available. Backup Real-Time Servers are configured using Real-Time Administrator and are available during sessions with Interaction Advisor clients implemented using the Java and .NET Clients.

Basic rights

The Real-Time Server-level privilege that entitles users to run Interaction Advisor client applications.

Begin Event function

A function you can define using Real-Time Studio to conduct certain processing whenever a particular event is triggered.

benefit

An arbitrary number indicating the benefit of an offer to your business if it is accepted. You can use, for example, a rating scale (such as 1 to 10) or a dollar value (such as profit). The greater the number, the greater the benefit. Benefit values are used during offer arbitration. You specify a benefit when you define an offer.

BestBenefit

The ranking strategy that ranks offers based on benefit. Also, the Real-Time Studio function that implements this strategy. Benefit is derived using a number you specify when you define an offer.

BestExpectedBenefit

The ranking strategy that ranks offers using the result obtained by multiplying benefit with likelihood. Also, the Real-Time Studio function that implements this strategy. See [benefit](#) and [likelihood](#).

cached profile

A profile that improves the performance of access to frequently used values by caching them and making them available using a key value across events.

campaign

See [Interaction Advisor campaign](#) and [making campaigns](#).

campaign activation

See [activation](#).

campaign building blocks

The campaign infrastructure that enables the use of Interaction Advisor to define Interaction Advisor campaigns. Campaign building blocks include campaign packages, real-time miners, marketing databases, customer events, profiles, and offer attributes.

campaign deployment

The activity performed using Real-Time Studio or Interaction Advisor that makes a campaign package available for use by Interaction Advisor client applications.

Campaign Editor permission

The permission that entitles Marketers to change the definitions of campaigns and delete them.

campaign execution

See [targeting](#).

campaign hierarchy

Two or more campaigns that influence each other. For example, when a campaign in a hierarchy is triggered, the activation and arbitration conditions of campaigns above it in the hierarchy are applied. And when a campaign in a hierarchy is triggered, campaigns below it in the hierarchy are also triggered.

campaign offer

See [offer](#).

Campaign Owner permission

The permission that entitles Marketers to define campaign level permissions as well as to change the definitions of and relationships among individual campaigns and offers.

campaign package

A file containing one or more Interaction Advisor campaigns. Campaign packages offer a way to subdivide campaigns into groups for separate campaign management, access, and Real-Time Server handling.

campaign package template

A campaign package containing predefined campaign elements that expedite creating new campaign packages. See [Customer profile](#), [System profile](#), [Now profile](#), [Today profile](#), [OfferTracker table](#), and [OfferAccepted event](#).

campaign package version

A rendition of a campaign package that is used for a particular purpose. A Test version is a deployed campaign package that is available for limited use in order to assess its effectiveness. A Production version is a deployed campaign package available for widespread use. You can define user groups to expedite access to Test and Production versions of campaign packages.

campaign targeting

See [targeting](#).

campaign testing

See [testing campaigns](#).

Catchphrase

A predefined static offer attribute that briefly summarizes an offer.

child campaign

A campaign that is defined below another campaign in a campaign hierarchy. The activation and arbitration conditions of the child's parent campaign influence the behavior and results of the child campaign when the child campaign is triggered.

clickstream data

Data that describes items of interest to a visitor using a web page, such as a particular web site area or particular topics or products.

clickstream event

A session event that delivers clickstream data for use in campaign targeting. Like other events, clickstream events can be sent directly from a client application.

client application

See [Interaction Advisor client application](#).

collaborative filtering

A technique used to derive items to recommend to an individual by using items that have been of interest to individuals with similar interests and characteristics. You can use recommendation models to implement collaborative filtering in a campaign package.

command file

See [Real-Time Server command file](#).

COM object

A Microsoft object. In a campaign package, you can create a COM object to access external information or use an external COM object as an event field.

component

An interface that lets you integrate Interaction Advisor campaign support into applications. See [Java Client](#), [.NET Client](#), [Socket Component](#), and [Web/HTTP Component](#).

composite field

A field of a real-time miner, recommender, or profile source that holds multiple values, such as products or counted items. Products are groups of items, and counted items are groups of items and a count indicating the number of times each item appears in the field.

conditions

Criteria you set up when defining campaigns and their offers that are used by Real-Time Server to match offers with customers and to control campaign availability and other campaign behavior. See [targeting criteria](#).

configuring Real-Time Server

The process of using Real-Time Administrator to set up Real-Time Server client communication attributes and campaign resource access. See [real-time marketing](#), startup parameters, and run-time parameters.

cost capacity

The offer characteristic describing the total amount you are willing to spend for offers accepted. It is the dollar threshold at which you want to stop extending an offer.

counted items

Items associated with composite fields that describe item values and a counter for each value.

customer event

A predefined activity in an Interaction Advisor client application that triggers the execution of one or more Interaction Advisor campaigns by Real-Time Server. When the event, such as a request for product information, occurs, the application sends an event message to Real-Time Server. The event message identifies the event name and usually contains information that uniquely identifies individual customers, such as an account number. See [session event](#), [clickstream event](#), [OfferAccepted event](#), and [default event](#).

Customer profile

The predefined profile that has an attribute which uniquely identifies a customer—CustomerId. This attribute is used to provide the customer identifier for OfferEvent table records. See [OfferTracker table](#).

Data Mart profile

A profile that makes Omni-channel Marketing information available for Interaction Advisor campaign targeting during a session. A Data Mart profile includes demographic and transaction information about customers whose data is stored in Omni-channel Marketing data mart. Data Mart profiles also include dynamic information about list memberships and treatment assignments.

date conditions

Targeting criteria based on calendar dates. Profile attributes associated with the Today profile are displayed when you use Interaction Advisor to construct date conditions.

default event

An event that returns a predefined offer to present to an individual when Real-Time Server becomes unavailable during campaign targeting. Client applications that use the Java or .NET Clients to implement Interaction Advisor campaign support can retrieve the predefined offer and store it for use when required.

deploying campaign packages

Making a campaign package and the campaigns in it available for use. Deploying campaign packages registers them with a particular Real-Time Server, making the package available for use by Interaction Advisor users or Interaction Advisor client applications that use that Real-Time Server or its slaves for campaign activities. You can deploy campaign packages from Real-Time Studio or from Interaction Advisor.

deployment

The activity that makes a campaign package available for use.

deployment directory

The directory under which deployed campaign packages and real-time miner resources are stored for use by a particular Real-Time Server. Path names to the default deployment directory, Services, are as follows:

- Windows: C:\Program Files\Epiphany\RT\Services
- Linux: *InstallDir/Services*

Campaign elements are stored in the following subdirectories within the Services directory:

- Real-time miners are stored in the *MdapEngine* subdirectory
- Campaign packages are stored in the *Targeting Engine* subdirectory

deployment group

A collection of Real-Time Servers, all of which use the same campaign packages and provide the same services to Interaction Advisor client applications. Deployment group members can be used as backup Real-Time Servers to other Real-Time Servers in the deployment group.

One Real-Time Server is designated the master, and the remaining ones are slaves. When one of the following elements of the master Real-Time Server changes, the slaves are automatically updated to remain in sync, including campaign packages and services they depend on and log events.

dynamic offer attribute

An element of information you can return to an Interaction Advisor client for an offer that is assigned a value at the time its campaign is targeted rather than at offer definition time. Use dynamic offer attributes to return recommendations to client applications after using a real-time miner or a recommendation model to derive recommendations. Contrast [static offer attribute](#).

dynamic profile attribute value

A value assigned to a profile attribute during campaign execution rather than during campaign definition. Dynamic attribute values can be used in static offer attribute values. See [profile](#).

End Event function

A function you can define using Real-Time Studio to conduct certain processing after all campaigns triggered by a particular event have been executed.

event field

A field associated with a customer event. An event message from an Interaction Advisor client application provides values for all fields associated with an event. Event field values provide information that can be used by source, profile, or other campaign components.

event startup function

See [Begin Event function](#).

event termination function

See [End Event function](#).

extended cost

The offer characteristic describing how much it costs to extend an offer.

failover

Backup. A Real-Time Server in a deployment group can be designated the failover, or backup, server to one or more other members of the same deployment group.

final arbitration function

A definition used in Real-Time Studio to set up cross-campaign arbitration conditions. Final arbitration conditions are used to determine which offers to return to an Interaction Advisor client application when customer events trigger multiple campaigns. Contrast [arbitration function](#) and [arbitration strategy](#).

function editor

The Real-Time Studio facility that is used to define and review campaign functions. See [functions](#).

functions

The Real-Time Studio definitions that are used to set up campaign targeting conditions and perform other campaign processing. See individual function names.

hierarchical campaign

See [campaign hierarchy](#).

HTTP session

Communication between an Interaction Advisor client implemented using the Web/HTTP, Java, or Socket Components and Real-Time Server. Use Real-Time Administrator to manage HTTP session parameters.

idle timeout

The number of minutes after the start of a session event that you want to terminate a session if it is not explicitly terminated first.

Interaction Advisor

The family of products you can license and integrate into your organization's own infrastructure. The system includes campaign definition and management applications, Real-Time Studio and Interaction Advisor, and campaign delivery components, RTCom client component and RTJava client, Real-Time Server, and Real-Time Administrator.

Interaction Advisor campaign

A collection of offers designed to achieve a particular business objective, such as increasing the sales of a specific product or improving a customer's wallet share. An Interaction Advisor campaign is executed, or targeted, when a customer event occurs, in real time, during interactions with customers or prospects, and it takes advantage of information about the individual that becomes available during that interaction. See [campaign targeting](#), [customer event](#), and [campaign hierarchy](#).

Interaction Advisor client application

An application into which Interaction Advisor campaign support has been integrated. Interaction Advisor clients trigger campaigns by spawning customer events and handle marketing offers that result. Interaction Advisor client applications host interactions with customers, such as call center and web site interactions.

Interaction Advisor JavaScript elements

JavaScript functions and variables that can be used in campaign functions, using Real-Time Studio, to manage Interaction Advisor campaign components and processing.

Java Adapter

The facility that lets you use Java objects from JavaScript functions in campaign packages.

Java Client

The component you use to integrate Interaction Advisor campaign support into Java applets and servlets.

JavaScript

The language used to write constructs in functions defined using Real-Time Studio and in Real-Time Server command files. You use a subset of the standard JavaScript language, with several special extensions, Interaction Advisor JavaScript functions and variables, that support Interaction Advisor campaign processing.

The JavaScript interpreter named ScriptEase is used to execute targeting criteria. The Real-Time JavaScript Reference, an online help system that is installed when you install Real-Time Studio, provides general information about ScriptEase JavaScript conventions as well as specific information about Real-Time campaign JavaScript objects.

JavaScript debugger

The ScriptEase Workshop program you can access from Real-Time Studio or Windows Explorer includes a JavaScript debugger. You can use this program to debug JavaScript in a campaign package function or a Real-Time Server command file.

keys

See [lookup key](#), [table key](#), and [cached profile](#).

likelihood

A number indicating the likelihood that a particular offer will be accepted by a customer. The higher the number, the greater the acceptance likelihood. Likelihood values, which can be used during offer arbitration, are derived by using likelihood conditions. See [likelihood conditions](#), [likelihood function](#), and [arbitration function](#).

likelihood conditions

Criteria Real-Time Server analyzes to derive a likelihood value for an offer. An offer's likelihood is derived by adjusting an initial likelihood value using influence factors you assign to profile attribute values. Initial likelihood values are computed using a real-time miner or other values you specify. Influence factors are logical expressions that increase or decrease the initial likelihood value to the extent you specify.

The likelihood value that results from application of likelihood conditions and can be used during offer arbitration.

likelihood function

The definition specified in Real-Time Studio to set up likelihood conditions for an offer. Likelihood functions return a numerical value. The greater the value, the greater the offer acceptance likelihood.

load balancer

A device that can be used to help manage the event traffic dispatched to individual Real-Time Servers. A load balancer forwards requests based on current Real-Time Server load and availability. Events from client applications implemented using the .NET, Java, Web/HTTP, and Socket Components can be dispatched using a hardware load balancer.

locking

The protection Real-Time Server provides for deployed campaign packages to prevent multiple users from changing a deployed package at the same time. Test or Production versions of campaign packages can be refined by only one person at a time, but you can browse through a campaign package another user has locked for refinement.

log directory

The directory in which Real-Time Server log files are stored, by default:

- Windows: C:\Program Files\Epiphan\RT\LogFiles
- Linux: *InstallDir*/LogFiles

log event

The logical name for a Real-Time Server log file, defined using Real-Time Administrator.

lookup key

One or more fields in lookup source records that provide the values needed to retrieve a specific record from the lookup source.

lookup sources

The campaign component you define using Real-Time Studio that is used to retrieve values used in functions, prediction sources, real-time miners, profiles, and other campaign components.

making campaigns

Defining an Interaction Advisor campaign using Real-Time Studio or Interaction Advisor. See [Interaction Advisor campaign](#).

mapped field

The field providing a value for a lookup source field, prediction source input field, real-time miner field, table column, profile attribute, or recommender field.

mapped object

A campaign element, event, function, and so on, that contains a field providing a value used in mapping. See mapping and mapped field.

mapping

Identifying the source of a value in fields associated with campaign elements such as profiles, recommenders, and other elements with fields.

margin

The offer characteristic describing the total profit realized when an offer is accepted.

Marketer rights

The Real-Time Server-level privilege intended primarily for individuals who define and manage Interaction Advisor campaigns, such as marketers and business managers. These individuals use Interaction Advisor to perform campaign management activities that they are entitled to, based on permissions that have been granted to them. See [permissions](#).

marketing databases

ODBC or Oracle tables or views that provide information used in lookup sources and prediction sources or receive information defined in sources that are tables.

master server

The Real-Time Server in a deployment group to which updated information, such as changed models or campaign packages, is made available. The master server propagates all changes to the slave servers in the deployment group so that all Real-Time Servers in the deployment group remain synchronized. See [deployment group](#).

model

A predictive model generated using a recommendation model. Recommendation models can be used to derive items to recommend to an individual.

monitoring campaigns

Reviewing information about campaign performance to determine how to refine existing campaigns or to decide whether new campaigns should be developed.

MostLikely

The ranking strategy that ranks offers using their acceptance likelihood. Also, the Real-Time Studio function that implements this strategy. See [likelihood](#).

normal profile

A profile containing values that are used only during a single event.

Now profile

The predefined profile that contains several attributes you can use to describe the time of day.

object references

See [COM object](#).

offer

A marketing message, resulting from Interaction Advisor campaign execution, that is sent to an Interaction Advisor client application for presentation to a customer when the offer's targeting criteria are satisfied. The offer may, for example, be a discount to entice customer loyalty, or it may be a pitch for a product that is likely to appeal to a customer.

When you define an offer, you define its content elements and its targeting criteria. Content elements constitute the information returned to Interaction Advisor clients when an offer is extended to a customer. Content elements include static and dynamic offer attributes. Targeting criteria provide campaign processing information for an offer, including qualification and likelihood conditions, offer benefit, real-time miner, and associated campaigns.

offer acceptance tracking

The automatic capabilities Real-Time Server provides to keep track of offers extended and accepted. See [OfferTracker table](#), [re-offer delay](#), and [real-time miner](#).

offer accepted cost

See [accepted cost](#).

OfferAccepted event

The predefined customer event that updates information about offers accepted. The event is designed to be triggered by an Interaction Advisor client application when a customer accepts an offer. The OfferAccepted event updates real-time miner information for the offer and any table used to track offers extended and accepted. OfferTracker is updated by default. See [OfferTracker table](#) and [real-time miner](#).

offer arbitration

See [arbitration](#).

offer attributes

The text and HTML values returned to Interaction Advisor clients for their use in presenting information for an offer. See [static offer attribute](#) and [dynamic offer attribute](#).

offer benefit

See [benefit](#).

offer cost capacity

See [cost capacity](#).

Offer Editor permission

The permission that entitles Marketers to change the definitions of offers and delete them.

offer extended cost

See [extended cost](#).

offer margin

See [margin](#).

Offer Owner permission

The permission that entitles Marketers to define offer level permission and change the definitions of offers.

offer qualification

See [qualification](#).

offer revenue

See [revenue](#).

OfferTracker table

The predefined table you can use to automatically keep track of offers extended and accepted. Real-Time Server writes a record to OfferTracker for each offer returned to a client and whenever the predefined event named OfferAccepted is received from a client to indicate offer acceptance.

offer tracking

The process of capturing information about the response to offers extended for use in refining real-time campaigns. See [automatic offer tracking](#).

Omni-channel Campaign Management

A campaign defined using Omni-channel Campaign Management. These can be deployed to Real-Time Server for real-time targeting.

Omni-channel Campaign Management attribute

A column within a Omni-channel Campaign Management dimension table. Customer attributes are used in Data Mart profiles.

Omni-channel campaign Management list

A list defined using Omni-channel Campaign Management that contains a group of individuals who satisfy particular selection criteria.

Package Deployer permission

The permission that entitles Marketers to deploy a production version of a campaign package.

Package Editor permission

The permission that entitles Marketers to create campaigns and offers in a campaign package as well as deploy a test version of the package.

Package Master rights

The Real-Time Server-level privilege that grants users complete control of all campaign packages deployed to a particular Real-Time Server.

Package Owner permission

The permission that entitles Marketers to conduct all the operations in Interaction Advisor for a package except deploying and copying it.

permissions

Privileges at the campaign package, campaign, and offer levels. Permissions regulate changes users with Marketer rights can make to a particular campaign package using Interaction Advisor. Permissions are defined using Real-Time Studio or Interaction Advisor. Marketer rights are defined using Real-Time Administrator.

port

The host port used to send requests to Real-Time Server. Eight ports are available for use: four administrative ports, for Real-Time Server, Interaction Advisor, and Real-Time Administrator requests and four event ports, primarily for Interaction Advisor client events. You configure the number, authentication, and timeout for each port using Real-Time Administrator.

Power e-Commerce Template

A campaign package template you can use in Real-Time Studio to expedite launching Interaction Advisor campaigns on the web.

profile

A user defined collection of items, known as profile attributes, that map data obtained from various information sources to elements needed for defining targeting criteria and other campaign logic. Profile attributes can also provide information to embed in static offer attributes so that these values are included in offer information returned to Interaction Advisor clients; see [dynamic profile attribute value](#).

Profiles offer a source independent way to refer to data obtained from campaign components such as customer events, lookup sources, functions, or other profiles. When the source of data for a profile attribute changes, campaign definitions using profile attributes to refer to the data need not change.

Kinds of profiles include [normal profile](#), [cached profile](#), [session profile](#), and [Data Mart profile](#)

profile attribute

An item associated with a profile.

profile attribute and field function

The Real-Time Studio definition you can use to assign a value to a profile attribute and other campaign components that have fields.

qualification

The process of evaluating qualification conditions to determine whether a customer is entitled to receive an offer.

qualification conditions

The criteria that define whether a customer is entitled to receive a particular offer. Qualification conditions consist of a [re-offer delay](#) and criteria that must evaluate to true. Criteria are characteristics that entitle a customer to receive an offer, such as having an account balance greater than a particular value or belonging to a particular demographic population. When the qualification conditions for an offer are satisfied for a particular customer, the customer is entitled to receive the offer. Whether the offer is actually extended to the customer depends on arbitration conditions.

qualification function

The definition specified in Real-Time Studio to set up qualification conditions for an offer. A qualification function returns true or false. When it returns true, its associated offer is a candidate for extending to a customer.

qualifying offer

An offer for which the qualification conditions are true.

ranking strategies

A strategy for ordering offers by their importance. When you set up arbitration conditions, you specify a strategy for ranking offers. You can use one of the default strategies or use Real-Time Studio to define a custom strategy. Default strategies are [MostLikely](#), [BestBenefit](#), and [BestExpectedBenefit](#).

Real-Time Administrator

The application used to configure Real-Time Server. Using Real-Time Administrator, you can manage startup and run-time parameters for any Real-Time Server on your network.

real-time marketing

Rapidly developing and deploying campaigns that respond to new market opportunities in days, not months; are as up to date as the latest information available; deliver offers tailored to individuals; and can be adjusted quickly and easily to increase their effectiveness and respond to changing markets.

real-time miner

A self-learning element that uses fields you specify to automatically refine campaign targeting for an offer every time the offer is accepted. The result is an increasingly accurate profile of individuals likely to accept a particular offer. After defining a real-time miner using Real-Time Studio, you associate it with one or more offers or with a campaign. You can use a real-time miner in offer likelihood conditions. See [likelihood conditions](#).

real-time miner field

One of the items associated with a real-time miner. The value for the field can be derived from a customer event, a lookup source, a profile, and so forth.

Real-Time Server

The application that executes Interaction Advisor campaigns, interacting with Interaction Advisor client applications to receive customer event information and send offer information. Real-Time Server is configured using Real-Time Administrator.

Real-Time Server command file

A file containing event and offer processing commands for Real-Time Server execution during interactions with clients implemented using the Socket or Web/HTTP Components. You can use the default command files **ProcessEvent.wpt** and **ProcessEvent.xml.wpt**, or you can create your own command files. Default command files are located in the following directories:

- Windows: C:\Program Files\Epiphany\RT\HTTP\WebPoint
- Linux: InstallDir/HTTPContent/WebPoint

Real-Time Studio

The application that provides complete campaign definition and management capabilities. Designed for the technical analyst, Real-Time Studio is most often used to set up campaign building blocks that facilitate campaign definition by marketing personnel using Interaction Advisor. Real-Time Studio can also be used to supplement, enhance, and test campaigns defined using Interaction Advisor as well as to manage campaigns without using Interaction Advisor. See [campaign building blocks](#).

recommendation model

A model that uses collaborative filtering techniques to derive recommendations.

RecommendRequest()

The Interaction Advisor JavaScript function used to construct a recommendation request object in preparation for deriving recommendations using a recommendation model.

refining campaigns

The process of changing campaign components to increase the rate of offer acceptance. Use Interaction Advisor monitoring capabilities as well as your own tables to monitor campaign results and identify ways to refine campaigns.

re-offer delay

The number of days, 365 by default, that must elapse between extending a particular offer and extending it again to the same customer. The re-offer delay is one of an offer's qualification conditions.

revenue

The offer characteristic describing the total revenue generated when an offer is accepted.

rights

Real-Time Server level privileges that define the applications an individual can use to interact with a particular Real-Time Server. The rights, from most to least powerful, are Administrator, Package Master, Marketer, and Basic.

RTDB

Real-Time Database. The database used to manage information for recommendation models, campaign packages, real-time miners, Data Mart profiles, and other data needed during Interaction Advisor campaign targeting. Audit trails describing campaign package changes can also be stored in the RTDB.

sample campaign packages

Examples that are installed along with Interaction Advisor software. **Golden.mop** features campaigns designed to promote bank products. The package named **comobj.mop** demonstrates how to use COM objects from a campaign package.

Secure Socket Layer

The communications option that safeguards interactions between client applications and Real-Time Server using authentication and encryption.

service name

A logical name by which various campaign components are referred to in campaign definitions: lookup sources, recommendation models, real-time miners, prediction sources, tables, and campaign packages. Service names insulate campaign component definitions from file name and location changes to their underlying physical resources. Service names are unique across all campaign resources associated with a particular Real-Time Server.

Service names are assigned when you register campaign components with individual Real-Time Servers—using Real-Time Administrator to set up services for recommendation models, lookup sources, tables, and real-time miners; and using Real-Time Studio to deploy campaign packages and real-time miners.

session

A series of contiguous events involving an individual and across which you want to accumulate information to characterize the context of the session.

session event

An event that triggers campaign processing you want to associate with a session, a series of contiguous events that occur during interactions with a particular individual.

session profile

A profile that accumulates data during the events associated with a session. See also Data Mart profile.

significance threshold

The number of offers that must be accepted before a real-time miner is used to compute offer acceptance likelihood during campaign targeting. The default is 20 offers. Set this threshold using Real-Time Administrator.

slave server

A Real-Time Server in a deployment group whose services are automatically updated whenever the master server resources change. See [master server](#) on page 94.

Socket Component

The component that provides a socket interface between applications and Real-Time Server. This component is of particular interest for applications that run on mainframe or specialized systems. Socket clients communicate with Real-Time Server using the HTTP protocol and socket connections to transmit events and receive offer information by invoking Real-Time Server command files.

sources

Campaign elements that provide information for use during campaign targeting. See [lookup sources](#), [real-time miner table](#), , and [stored procedures](#).

static offer attribute

An element of information you can return to an Interaction Advisor client for an offer that is assigned a value when the offer is defined. Contrast [dynamic offer attribute](#).

stored procedures

The campaign component you define using Real-Time Studio that enables campaigns to take advantage of the processing and information available when executing Informix and Oracle stored procedures and stored functions.

System profile

The predefined profile that has several attributes describing time, such as **CurrentDay**.

table

An Interaction Advisor campaign element that is used to record information generated during campaign processing, such as campaign results. Tables are write-only data sources in ODBC or Oracle databases. Read operations on tables are not supported. To retrieve data from a table, define a lookup source based on its underlying data source. A predefined table, OfferTracker, supports automated offer tracking. See [OfferTracker table](#).

table key

The column value or values that uniquely identify records in a table. See [table](#).

targeting

The processing associated with campaign execution. A campaign is targeted, or executed, when one of its related events occurs and the campaign is active at the time. During campaign targeting, targeting criteria for all campaigns associated with the event are used to determine whether to send one or more offers back to the Interaction Advisor client application for presentation to a customer. Contrast with [triggering](#).

targeting criteria

The conditions that determine which offer or offers to forward to an Interaction Advisor client application when a campaign is targeted. Targeting criteria consist of offer-level criteria, campaign-level criteria, and package-level criteria. Offer-level criteria include qualification conditions, likelihood conditions, and offer benefit. Campaign-level criteria include activation and arbitration conditions. Package-level criteria include final arbitration functions.

template

Real-Time Manager includes campaign package templates, which contain predefined items that are usually part of a campaign package. See the topic “Campaign Package Templates” in Real-Time Studio Online Help.

test deployment

Providing limited access to a campaign package prior to full, or production, deployment to ensure that its campaigns are sufficiently effective for widespread use.

testing campaigns

Conducting activities that ensure a campaign performs as expected and that correct Interaction Advisor client-Real-Time Server connections are established. Real-Time Studio provides several test capabilities, ranging from event simulation to JavaScript debugging, to test individual campaign packages. Interaction Advisor provides ways to analyze dependencies and conflicts within campaigns.

To assess campaign effectiveness under run-time conditions, you can define user test groups using Real-Time Administrator and monitor campaign performance using Interaction Advisor.

time conditions

Targeting criteria based on time of day. Profile attributes associated with the Now profile are displayed when you use Interaction Advisor to construct time conditions.

time zone adjustment

The adjustment that must be made to Greenwich Mean Time to represent local time.

Today profile

The predefined profile that contains several attributes you can use to describe days of the year.

touchpoint

An application used during interactions with customers and prospects, for example, a call center application or a web server application.

treatment

A code that indicates the method by which selected members of a Omni-channel campaign Management campaign are to be contacted. When a Omni-channel campaign Management campaign is deployed to Real-Time Server, each of its treatments becomes an offer in an Interaction Advisor campaign.

triggering

The action taken when a customer event occurs to start processing of the event's associated campaign or campaigns. Contrast with [targeting](#).

user access group

A collection of registered users who have been assigned Marketer rights for a particular Real-Time Server. User access groups facilitate assigning permissions to Marketers.

user package group

A collection of users configured to use the same campaign package by default when an Interaction Advisor client sends an event that has no campaign package name associated with it. Use Real-Time Administrator to set up user groups and assign individual users to them.

utility function

A Real-Time Studio definition used to perform miscellaneous campaign processing using JavaScript.

validation

The process of ensuring that profile conditions and offer attribute values constructed using Interaction Advisor are correct. To implement validation, use Real-Time Studio to write JavaScript functions that validate values the Interaction Advisor user enters. These entries are for profile attributes or static offer attributes and functions that display valid values for Interaction Advisor user selection.

version

See [campaign package version](#).

Web/HTTP Component

The component that enables you to integrate Interaction Advisor campaigns into web server applications. You embed URLs that establish a Real-Time Server connection and invoke campaign management operations in Real-Time Server command files. This component can be used with any programming language that supports the HTTP protocol.

Wrapup Session function

The package-level JavaScript function that performs any processing required when a session terminates.

Index

Special Characters

.NET client [17](#)
client socket timeout, overriding [17](#)

A

access groups [42](#)
accessing [24](#), [45](#), [50](#), [85](#)
 BIO data [50](#)
 data [45](#)
 deployment group settings [24](#)
 JavaScript debugging settings [85](#)
Active Directory Service user ID [39](#)
adding [24](#), [27](#), [74](#)
 events to custom logs [74](#)
 slave to a deployment group [27](#)
 slave with real-time-miner learnings [27](#)
 users to groups [24](#)
adding to groups [24](#)
administrator rights [38](#)
and Real-Time Miners [56](#)
anonymous users [32](#)
anonymous, for port requests [32](#)
ASCII log files [72](#)
assigning a slave as a master [28](#)
assigning as a master [28](#)
associating masters and slaves [25](#)
associating slaves [25](#)
associating with a master [25](#)
audit trails [17](#)
 See package audit trails. [17](#)
authentication [32](#), [39](#)
 for port requests [32](#)
 user settings [39](#)

B

backup servers [17](#), [23](#), [25](#), [40](#)
 for single sign-on [17](#)
 port settings [25](#)
basic rights [38](#)
BIO Access [50](#)
BIO Access service [50](#)
BIO data [50](#)

C

cal [85](#)
Campaign package deployment tools [66](#)
 DeployPackage [66](#)
 GetPackage [66](#)
campaign packages [65](#)

campaign packages— [64](#)
campaign permissions [38](#)
CAMPAIGNPACKAGE ERROR event [72](#)
CAMPAIGNPACKAGE ERROR log event [72](#)
certificate for testing [35](#)
certificates [35–36](#)
 in server.pem [35](#)
 making a certificate request [36](#)
changing port interface settings [34](#)
changing RT Server settings [14](#)
changing settings [34](#)
classpath setting [54](#)
 separators for multiple .jar files [54](#)
client setup connectivity [83](#)
client socket timeout [17](#)
client socket timeout, overriding [17](#)
command line options [80](#)
Command Thread Pool Size [17](#)
communication via HTTP [31](#)
communication with other components [31](#)
configuring [35](#)
configuring security options [35](#)
contacting Customer Support [9](#)
 Customer Support [9](#)
content base directory [31](#)
convergence threshold for Real-Time Miners [56](#)
creating [25](#)
creating a log record [74](#)
creating a service [62](#)
creating Real-Time Miner services [58](#)
custom logs, defining [74](#)
Customer Support [9](#)

D

data [45](#)
data access [35](#), [45–46](#), [82](#)
 modifying a service [35](#)
 settings [46](#)
 troubleshooting [82](#)
 user name requirements [82](#)
data access service [35](#)
DATA_ACCESS_TRACE [73](#)
DATA_ACCESS_TRACE log [73](#)
Database [38](#), [45](#)
database permissions [83](#)
debugging [85](#)
debugging settings [85](#)
default package assignments [43](#)
default port assignments [32](#)
defining [23](#), [25](#), [40](#), [43](#)
 deployment groups [25](#)
 registered users [23](#), [40](#)

defining (*continued*)
 user package groups [25](#), [43](#)
defining custom logs [74](#)
deleting [27](#), [59](#), [75](#)
 log event [75](#)
 log event column [75](#)
 Real-Time Miner services [59](#)
deployment database service [17](#)
deployment directory [17](#)
deployment group settings [24](#)
deployment groups [23–25](#), [27–29](#), [38](#), [56](#), [60](#)
 and Real-Time Miners [56](#)
 assigning a slave as a master [28](#)
 associating masters and slaves [25](#)
 defining [25](#)
 group members setting [25](#)
 master server, designating [25](#), [38](#)
 master-slave relationships [23](#)
 Real-Time miner definitions across [60](#)
 reassigning the master role [27](#)
 server role [25](#)
 settings [24](#)
 slave servers, designating [25](#)
 splitting [29](#)
DeployPackage [66](#)
designating [25](#)
designating a master server [25](#), [38](#)
 See Also slave servers. [25](#), [38](#)
designating a slave server [25](#)
directories [17](#)
disabling SSL [35](#)
dmperfux [80](#)
 command line options [80](#)

E

editor permissions [38](#)
effect of creating real-time miner on [60](#)
enabling [17](#), [31](#), [35](#), [85](#)
 HTTP logging [31](#)
 remote debugging [85](#)
 single sign-on [17](#)
 SSL [35](#)
enabling logging [31](#)
EVENT_TIME [74](#)
EVENT_TIME_SEC [74](#)
events queued in RT Server, number of [17](#)
events to custom logs [74](#)

F

failover servers [17](#), [23](#), [25](#), [40](#)
 for single sign-on [17](#)
 port settings [25](#)
for communication between components [35](#)
for master and slave servers [23](#)
for port requests [32](#)

for recommendation models [62](#)
for single sign-on [17](#)

G

GetPackage [66](#)
granting access to packages, campaigns and offers [38](#)
group members setting [25](#)
group members, of deployment group [25](#)
groups [38](#), [42](#)
 granting access to packages, campaigns and offers [38](#)
 user access groups [42](#)

H

HTTP [31](#)
 content base directory [31](#)
 enabling logging [31](#)
 log file [31](#)
 settings [31](#)
HTTP logging [31](#)

I

ID [39](#)
in deployment groups [56](#), [60](#)
in server.pem [35](#)
information about users [39](#)

J

Java adapter [54](#)
 settings [54](#)
Java Client [17](#)
 client socket timeout, overriding [17](#)
JavaScript debugger [85](#)
JavaScript debugging settings [85](#)

L

le [85](#)
log event [75](#)
log event column [75](#)
log file [31](#)
log file for HTTP [31](#)
logging [72–75](#)
 CAMPAIGNPACKAGE ERROR event [72](#)
 creating a log record [74](#)
 DATA_ACCESS_TRACE [73](#)
 defining custom logs [74](#)
 modifying logs— [75](#)
 settings [73](#)
logging on to RT Administrator for the first time [12](#)

M

making a certificate request [36](#)
 managing services [58](#)
 marketer rights [38](#)
 master port setting, for slave [25](#)
 master server [25](#), [28](#), [38](#), [60](#)
 assigning a slave as a master [28](#)
 associating slaves [25](#)
 designating a master server [25](#), [38](#)
 See Also slave servers. [25](#), [38](#)
 effect of creating real-time miner on [60](#)
 settings [25](#)
 master server settings [25](#)
 master server, designating [25](#), [38](#)
 master-slave relationships [23](#), [25](#)
 creating [25](#)
 maximum database connections [46](#)
 maximum events queued [17](#)
 maximum sleep/work ratio, for recommendation models [62](#)
 MdapEngine subdirectory [17](#)
 minimum database connections [46](#)
 minimum percent change to process, for recommendation models [62](#)
 model breadth, for recommendation models [62](#)
 model builder setting [62](#)
 modifying [35](#), [43](#), [58](#), [75](#)
 data access service [35](#)
 default package assignments [43](#)
 log event [75](#)
 Real-Time Miner services [58](#)
 modifying a service [35](#)
 modifying assignments [43](#)
 modifying logs– [75](#)
 moperror.log [72](#)
 moving campaign packages to another Real-Time Server [20](#)
 multimedia file access [83](#)
 multiple .jar files [54](#)
 multiple servers [22](#)

O

offer permissions [38](#)
 older Real-Time Miners, storing and sharing [60](#)
 overriding the client socket timeout [17](#)
 overview for RT Server [14](#)
 overview of RT Server settings [14](#)
 owner permissions [38](#)
 owner, editor and review [38](#)

P

package audit trails [17](#), [19](#)
 saving [19](#)
 storing in RTDB [17](#)

package deployer [38](#)
 package deployer permissions [38](#)
 package groups [42](#)
 package master rights [38](#)
 package permissions [38](#)
 packages [19](#), [38](#), [43](#)
 modifying assignments [43](#)
 permissions for [38](#)
 saving [19](#)
 performance counters [77](#)
 performance monitoring utility [80](#)
 permissions for [38](#)
 permissions– [25](#), [38](#)
 port assignments [32](#)
 port interface [32](#), [34](#)
 changing settings [34](#)
 settings [32](#)
 port interface timeout [32](#)
 port request authentication [32](#)
 port settings [25](#)
 ports, default assignments [32](#)
 printing [10](#)
 private.pem [36](#)
 product documentation [10](#)
 printing [10](#)
 production and test packages [64](#)

Q

qualified user ID [39](#)

R

Real-Time Administrator [8](#), [12–13](#)
 starting– [12–13](#)
 username and password [12](#)
 Real-Time miner definitions across [60](#)
 real-time miner learnings [27](#)
 Real-Time Miner services [58–59](#)
 Real-Time Miners [17](#), [19](#), [56](#), [58–60](#)
 in deployment groups [56](#), [60](#)
 managing services [58](#)
 renaming or deleting elements [56](#)
 saving [19](#)
 sharing among RT Servers [59](#)
 storing and sharing older miners [60](#)
 storing models in database [17](#)
 Real-Time Server [12](#), [14](#), [17](#), [22–23](#), [25](#), [31–32](#), [35](#), [40](#), [43](#), [45](#), [59](#), [72](#), [77](#), [82](#)
 backup servers [23](#), [40](#)
 communication with other components [31](#)
 data access [45](#)
 directories [17](#)
 logging [72](#)
 master server settings [25](#)
 master-slave relationships [23](#)
 multiple servers [22](#)

Real-Time Server (*continued*)
performance counters [77](#)
port assignments [32](#)
settings [17](#)
settings overview [14](#)
settings, changing [14](#)
sharing Real-Time Miners [59](#)
slave server settings [25](#), [43](#)
starting, in Windows [12](#)
troubleshooting [82](#)
using SSL [35](#)
viewing settings [14](#)
reassigning storage to another Real-Time Server [20](#)
reassigning the master role [27](#)
recommendation models [62](#)
recommendation models– [62](#)
registered users [23](#), [40](#)
remote debugging [85](#)
removing [43](#)
removing a slave from a deployment group [29](#)
removing user groups [43](#)
renaming or deleting elements [56](#)
resource storage [17](#)
restoring lost data [20](#)
reviewer permissions [38](#)
rights [38](#)
rights, privileges and permissions [38](#), [45](#)
rpttrace.log [73](#)

S

saving [19](#)
saving packages [19](#)
saving Real-Time Miners [19](#)
security [35](#)
certificates [35](#)
configuring [35](#)
SSL. See SSL (secure sockets layer). [35](#)
See Also slave servers. [25](#), [38](#)
See package audit trails. [17](#)
separators [54](#)
multiple .jar files [54](#)
separators for multiple .jar files [54](#)
server role [25](#)
server socket read timeout [17](#)
server.pem [35](#)
service guarantee [32](#)
service guarantee timeout [32](#)
services [17](#), [20](#), [50](#)
BIO Access [50](#)
deployment directory [17](#)
restoring lost data [20](#)
services deployment directory [17](#), [20](#)
setting the classpath [54](#)
settings [14](#), [17](#), [24–25](#), [31–32](#), [39](#), [43](#), [46](#), [62](#), [65](#), [73](#)
campaign packages [65](#)
deployment groups [24](#)
overview for RT Server [14](#)

settings (*continued*)
Real-Time Server [17](#)
recommendation models [62](#)
settings overview [14](#)
settings–?? [54](#)
settings, changing [14](#)
sharing among RT Servers [59](#)
sharing Real-Time Miners [59](#)
significance threshold for Real-Time Miners [56](#)
single sign-on [17](#)
single sign-on, enabling [17](#)
slave server settings [25](#), [43](#)
slave servers [25](#), [28](#), [43](#)
assigning as a master [28](#)
associating with a master [25](#)
designating [25](#)
settings [25](#), [43](#)
slave servers, designating [25](#)
slave to a deployment group [27](#)
slave with real-time-miner learnings [27](#)
socket [17](#)
Socket Thread Pool Size [17](#)
splitting [29](#)
splitting deployment groups [29](#)
SSL [35](#)
SSL (secure sockets layer) [35–36](#)
certificate for testing [35](#)
certificates [35](#)
enabling [35](#)
for communication between components [35](#)
making a certificate request [36](#)
SSL. See SSL (secure sockets layer). [35](#)
starting RT Administrator– [12–13](#)
starting RT Server in Windows [12](#)
starting– [12–13](#)
starting, in Windows [12](#)
store campaign packages in database [17](#)
store package audit trails in database [17](#)
storing and sharing older miners [60](#)
storing in RTDB [17](#)
storing models in database [17](#)
storing RT Miner models in database [17](#)
surprise factor, for recommendation models [62](#)
system_trace.log [73](#)

T

TargetingEngine subdirectory [17](#)
test and production packages [64](#)
Thawte security certificates [35](#)
threads [62](#)
for recommendation models [62](#)
timeouts [17](#), [32](#)
service guarantee [32](#)
socket [17](#)
troubleshooting [82](#)
troubleshooting deployment [82](#)
troubleshooting– [82–83](#)

U

- user [39](#)
 - ID [39](#)
- user access groups [42](#)
- user ID [39](#)
- user name requirements [82](#)
- user package groups [25](#), [42–43](#)
 - defining [25](#), [43](#)
 - removing [43](#)
- user rights [23](#), [38](#)
 - for master and slave servers [23](#)
- user settings [39](#)
- username and password [12](#)
- users [23–25](#), [32](#), [38–40](#), [42](#), [45](#)
 - access groups [42](#)
 - adding to groups [24](#)
 - anonymous, for port requests [32](#)
 - authentication [39](#)
 - defining [23](#), [40](#)
 - package groups [42](#)

- users (*continued*)
 - permissions— [25](#), [38](#)
 - rights [38](#)
 - rights, privileges and permissions [38](#), [45](#)
 - settings [39](#)
- users to groups [24](#)
- using java [54](#)
- using multiple Real-Time Servers [8](#)
- using SSL [35](#)

V

- VeriSign [35](#)
- viewing settings [14](#)

W

- web browser, starting RT Administrator from [13](#)
- Windows authentication [39](#)
 - user ID [39](#)