

WebFOCUS

ReportCaster and Two-Way Email
API for Self-Service Applications
Version 4 Release 3.6

Cactus, EDA/SQL, FIDEL, FOCCALC, FOCUS, FOCUS Fusion, FOCUS Vision, Hospital-Trac, Information Builders, the Information Builders logo, Parlay, PC/FOCUS, SmartMart, SmartMode, SNAPpack, TableTalk, WALDO, Web390, WebFOCUS and WorldMART are registered trademarks and EDA, iWay, and iWay Software are trademarks of Information Builders, Inc.

Acrobat and Adobe are registered trademarks of Adobe Systems Incorporated.

Allaire and JRun are trademarks of Allaire Corporation.

UniVerse is a registered trademark of Ardent Software, Inc.

AvantGo is a trademark of AvantGo, Inc.

WebLogic is a registered trademark of BEA Systems, Inc.

SUPRA and TOTAL are registered trademarks of Cincom Systems, Inc.

Alpha, DEC, DECnet, and NonStop are registered trademarks and Tru64, OpenVMS, and VMS are trademarks of Compaq Computer Corporation.

CA-ACF2, CA-Datcom, CA-IDMS, CA-Top Secret, and Ingres are registered trademarks of Computer Associates International, Inc.

MODEL 204 and M204 are registered trademarks of Computer Corporation of America.

HP MPE/iX is a registered trademark of Hewlett Packard Corporation.

Informix is a registered trademark of Informix Software, Inc.

Intel is a registered trademark of Intel Corporation.

ACF/VTAM, AIX, AS/400, CICS, DB2, DRDA, Distributed Relational Database Architecture, IBM, MQSeries, MVS, OS/2, OS/390, OS/400, RACF, RS/6000, S/390, VisualAge, VM/ESA, VTAM, and WebSphere are registered trademarks and DB2/2, HiperSpace, IMS, MVS/ESA, QMF, SQL/DS, and VM/XA are trademarks of International Business Machines Corporation.

INTERSOLVE and Q+E are registered trademarks of INTERSOLVE.

Orbit is a registered trademark of Iona Technologies Inc.

Approach and DataLens are registered trademarks of Lotus Development Corporation.

ActiveX, FrontPage, Microsoft, MS-DOS, PowerPoint, Visual Basic, Visual C++, Visual FoxPro, Windows, and Windows NT are registered trademarks of Microsoft Corporation.

Teradata is a registered trademark of NCR International, Inc.

Netscape, Netscape FastTrack Server, and Netscape Navigator are registered trademarks of Netscape Communications Corporation.

NetWare and Novell are registered trademarks of Novell, Inc.

Oracle is a registered trademark and Rdb is a trademark of Oracle Corporation.

Palm is a trademark and Palm OS is a registered trademark of Palm Inc.

INFOAccess is a trademark of Pioneer Systems, Inc.

Progress is a registered trademark of Progress Software Corporation.

BlackBerry is a trademark and RIM is a registered trademark of Research In Motion Limited.

Red Brick Warehouse is a trademark of Red Brick Systems.

SAP and SAP R/3 are registered trademarks and SAP Business Information Warehouse and SAP BW are trademarks of SAP AG.

Silverstream is a trademark of Silverstream Software.

ADABAS is a registered trademark of Software A.G.

CONNECT:Direct is a trademark of Sterling Commerce.

Java and all Java-based marks, NetDynamics, Solaris, SunOS, and iPlanet are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

PowerBuilder, Powersoft, and Sybase are registered trademarks and SQL Server is a trademark of Sybase, Inc.

Unicode is a trademark of Unicode, Inc.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Due to the nature of this material, this document refers to numerous hardware and software products by their trade names. In most, if not all cases, these designations are claimed as trademarks or registered trademarks by their respective companies. It is not this publisher's intent to use any of these names generically. The reader is therefore cautioned to investigate all claimed trademark rights before using any of these names other than to refer to the product described.

Copyright © 2002, by Information Builders, Inc. All rights reserved. This manual, or parts thereof, may not be reproduced in any form without the written permission of Information Builders, Inc.

Printed in the U.S.A.

Preface

This documentation describes how to use the ReportCaster and Two-Way Email Application Programming Interface (API), Version 4 Release 3.6. These APIs use JavaServer Pages™ (JSP™) technology, a Java™ servlet, or a Java application with JavaBeans™ components to create independent, customized environments that offer the features of ReportCaster and Two-Way Email. This manual is intended for developers of self-service applications.

How This Document Is Organized

This manual includes the following chapters and appendices:

Chapter/Appendix		Contents
1	<i>Introducing the ReportCaster API</i>	Introduces the ReportCaster API, which includes ReportCaster Bean API, ReportCaster Servlet API, and ReportCaster API subroutines.
2	<i>ReportCaster Bean API</i>	Describes the ReportCaster Bean API, which should be used for scheduling and log functions.
3	<i>ReportCaster Servlet API</i>	Describes the ReportCaster Servlet API, which should be used for Address Book functionality.
4	<i>ReportCaster API Subroutines</i>	Describes the ReportCaster API subroutines, which are a set of C-based API functions that call a servlet from a procedure.
5	<i>ReportCaster API JSP Samples</i>	Describes the ReportCaster API JavaServer Pages (JSP) sample forms, which use Bean functionality. Use these samples or customize them for your application needs.
6	<i>ReportCaster API Servlet Samples</i>	Describes the ReportCaster API Servlet samples, which are a full set of HTML sample forms that use servlet functionality. Use these samples or customize them for your application needs.

Chapter/Appendix		Contents
7	<i>ReportCaster Repository Reports</i>	Describes reports that access the ReportCaster Repository that are available from the ReportCaster API sample page.
8	<i>Self-Service Two-Way Email Overview</i>	Provides an overview of Self-Service Two-Way Email.
9	<i>Self-Service Two-Way Email Samples</i>	Describes the sample interface that can be used by developers to create a Self-Service Two-Way Email template.
10	<i>Self-Service Two-Way Email Administrator Console</i>	Describes the Self-Service Two-Way Email Administrator Console.
A	<i>ReportCaster API Debugging Techniques</i>	Discusses the tools and techniques available for debugging your ReportCaster API application. Includes information about tracing, examining a query string, and using the API Query Tool.
B	<i>ReportCaster API Messages and Codes</i>	Provides the return codes and messages that indicate whether a ReportCaster subroutine executed successfully.
C	<i>API-based MRE User</i>	Explains how you can use the API-based MRE User type to maintain server schedules and view log reports for jobs created using the ReportCaster API.

Documentation Conventions

The following conventions apply throughout this manual:

Convention	Description
<code>THIS TYPEFACE</code> or <code>this typeface</code>	Denotes syntax that you must enter exactly as shown.
<i>this typeface</i>	Represents a placeholder (or variable) in syntax for a value that you or the system must supply.
<u>underscore</u>	Indicates a default setting.
<i>this typeface</i>	Represents a placeholder (or variable) in a text paragraph, indicates a cross-reference, or emphasizes an important term.
this typeface	Highlights file names and commands (in a text paragraph) that must be lowercase.
this typeface	Indicates buttons, menu items, and dialog box options you can click or select.
Key + Key	Indicates keys that must be pressed simultaneously.
{ }	Indicates two choices from which you must choose one. You type one of these choices, not the braces.
[]	Indicates a group of optional parameters. None are required, but you may select one of them. Type only the information within the brackets, not the brackets.
	Separates two mutually exclusive choices in a syntax line. You type one of these choices, not the symbol.
...	Indicates that you can enter a parameter multiple times. Type only the parameters, not the ellipsis points (...).
. . .	Indicates that there are (or could be) intervening or additional commands.

Related Publications

See the Information Builders Publications Catalog for the most up-to-date listing and prices of technical publications, plus ordering information. To obtain a catalog, contact the Publications Order Department at (800) 969-4636.

You can also visit our World Wide Web site, <http://www.informationbuilders.com>, to view a current listing of our publications and to place an order.

Customer Support

Do you have questions about the ReportCaster and Two-Way Email API?

Call Information Builders Customer Support Services (CSS) at (800) 736-6130 or (212) 736-6130. Customer Support Consultants are available Monday through Friday between 8:00 a.m. and 8:00 p.m. EST to address all your WebFOCUS questions. Information Builders consultants can also give you general guidance regarding product capabilities and documentation. Please be ready to provide your six-digit site code number (xxxx.xx) when you call.

You can also access support services electronically, 24 hours a day, with InfoResponse Online. InfoResponse Online is accessible through our World Wide Web site, <http://www.informationbuilders.com>. It connects you to the tracking system and known-problem database at the Information Builders support center. Registered users can open, update, and view the status of cases in the tracking system, and read descriptions of reported software issues. New users can register immediately for this service. The technical support section of www.ibi.com also provides usage techniques, diagnostic tips, and answers to frequently asked questions.

To learn about the full range of available support services, ask your Information Builders representative about InfoResponse, or call (800) 969-INFO.

Information You Should Have

To help our consultants answer your questions most effectively, be ready to provide the following information when you call:

- Your six-digit site code number (xxxx.xx).
- Your WebFOCUS configuration:
 - The front end you are using, including vendor and release.
 - The communications protocol (for example, TCP/IP or HLLAPI), including vendor and release.
 - The software release.
 - The server you are accessing, including release (for example, 4.3.6).

- The stored procedure (preferably with line numbers) or FOCUS commands being used in server access.
- The name of the Master File and Access File.
- The exact nature of the problem:
 - Are the results or the format incorrect; are the text or calculations missing or misplaced?
 - The error message and return code, if applicable.
 - Is this related to any other problem?
- Has the procedure or query ever worked in its present form? Has it been changed recently? How often does the problem occur?
- What release of the operating system are you using? Has it, WebFOCUS, your security system, communications protocol, or front-end software changed?
- Is this problem reproducible? If so, how?
- Have you tried to reproduce your problem in the simplest form possible? For example, if you are having problems joining two databases, have you tried executing a query containing just the code to access the database?
- Do you have a trace file?
- How is the problem affecting your business? Is it halting development or production? Do you just have questions about functionality or documentation?

User Feedback

In an effort to produce effective documentation, the Documentation Services staff at Information Builders welcomes any opinion you can offer regarding this manual. Please use the Reader Comments form at the end of this manual to relay suggestions for improving the publication or to alert us to corrections. You can also use the Document Enhancement Request Form on our Web site, <http://www.informationbuilders.com>.

Thank you, in advance, for your comments.

Information Builders Consulting and Training

Interested in training? Information Builders Education Department offers a wide variety of training courses for this and other Information Builders products.

For information on course descriptions, locations, and dates or to register for classes, visit our World Wide Web site (<http://www.informationbuilders.com>) or call (800) 969-INFO to speak to an Education Representative.

Contents

1	Introducing the ReportCaster API	1-1
	Terminology for the ReportCaster API	1-2
	Installation Requirements	1-2
	ReportCaster Bean API Overview	1-3
	Scheduling Options	1-3
	Log File Options.....	1-4
	ReportCaster Servlet API Overview.....	1-4
	ReportCaster API Subroutines Overview	1-4
2	ReportCaster Bean API	2-1
	Beans Within the ibi.broker.beans Package	2-2
	Beans Within the ibi.broker.beans.handler Package.....	2-2
	ReportCaster Repository Access	2-3
	Security for the ReportCaster Bean API.....	2-6
	User Validation	2-6
	Configuration Information	2-6
	DSTAuthenticate Bean Properties.....	2-7
	Using the ReportCaster Bean API	2-8
	DTDs for ReportCaster XML Result Sets	2-12
	Processing ReportCaster Bean API Result Sets	2-18
	Retrieving Column Data.....	2-18
	Obtaining the Number of Rows of Data.....	2-19
	Using Property Names to Retrieve Column Data.....	2-20
	JavaServer Pages Technology	2-21
	Calling a JSP From an HTML Form or URL.....	2-22
	Using a JSP to Set Bean Properties	2-23
3	ReportCaster Servlet API.....	3-1
	ReportCaster Servlet API Descriptions	3-2
	ReportCaster Repository Tables Accessed by the Servlets	3-3
	Servlet Security	3-6
	IBIB_userid Parameter.....	3-7
	Calling a Servlet From an HTML Form	3-8
	Servlet Parameters.....	3-9
	Sample HTML Forms	3-9

Maintaining and Displaying a Distribution List With a Servlet	3-10
Maintaining Distribution Lists Using the DSTDLBULK Servlet	3-10
DSTDLBULK Parameters	3-11
Creating an External File.....	3-15
Generating Text Box Input.....	3-16
Creating and Populating a Distribution List Using Text Box Input	3-16
Maintaining Single Distribution List Members Using the DSTDLMEM Servlet	3-17
DSTDLMEM Parameters	3-17
Displaying a Distribution List Using the DSTDLLIST Servlet.....	3-19
DSTDLLIST Parameters	3-19
Scheduling a Job Using the DSTSCHEd Servlet.....	3-20
DSTSCHEd Requirements	3-20
DSTSCHEd Parameters.....	3-21
Mail Parameters for DSTSCHEd	3-28
FTP Parameters for DSTSCHEd	3-30
Notification Parameters for DSTSCHEd.....	3-31
Passing Values to a Scheduled Job Using DSTSCHEd.....	3-33
Setting the Status of a Job Using the DSTACTIVE Servlet.....	3-35
DSTACTIVE Parameters	3-35
Immediately Running a Scheduled Job Using the DSTRUNNOW Servlet	3-37
DSTRUNNOW Parameters.....	3-38
Displaying Log Information Using the DSTLOG Servlet.....	3-40
DSTLOG Parameters	3-40
Log Content.....	3-43
4 ReportCaster API Subroutines	4-1
API Subroutines	4-2
Tables Accessed by the Subroutines.....	4-2
Subroutine Security	4-3
Calling a Subroutine From a Procedure.....	4-3
Calling an API Subroutine Using -SET.....	4-4
Maintaining Distribution Lists Using the DSTBULK Subroutine.....	4-8
DSTBULK Subroutine Arguments	4-8
Maintaining Single Distribution List Members Using the DSTMEM Subroutine	4-18
DSTMEM Subroutine Arguments.....	4-18
Running a Scheduled Job Using the DSTRUN Subroutine.....	4-25
DSTRUN Subroutine Arguments.....	4-26
Using Amper Variables Within a Subroutine	4-34

5	ReportCaster API JSP Samples	5-1
	Scheduling Functions	5-2
	Log Functions.....	5-3
	Logging on to the ReportCaster Bean API.....	5-4
	JSP Samples	5-6
	JSP Schedule Options.....	5-7
	Schedule List.....	5-13
	JSP Reports	5-16
6	ReportCaster API Servlet Samples	6-1
	Sample HTML Calling Forms.....	6-2
	Logging on to the ReportCaster Servlet API.....	6-3
	Servlet Samples	6-5
	Servlet Schedule Options	6-5
	Servlet Distribution Options.....	6-14
	Servlet Report Options	6-20
7	ReportCaster Repository Reports	7-1
	Running ReportCaster Repository Reports	7-2
	Relational Databases	7-3
	Distribution Type Report.....	7-5
	Address Book Report	7-7
	Schedule Contacts Report.....	7-7
	No Contacts Report	7-8
	ReportCaster Users Report	7-8
	Alerts Schedules Report	7-9
8	Self-Service Two-Way Email Overview	8-1
	Installation Requirements	8-2
	Using the Self-Service Two-Way Email Beans.....	8-2
	Self-Service Two-Way Email Beans Within the ibi.broker.beans and ibi.broker.beans.handler Packages	8-3
	Security for Self-Service Two-Way Email.....	8-7
	Processing Overview Using JSP Technology.....	8-7
	Processing Overview for a Self-Service Two-Way Email Request.....	8-8
	Self-Service Two-Way Email Interface.....	8-9
	Self-Service Two-Way Email Administration.....	8-9

9	Self-Service Two-Way Email Samples	9-1
	Self-Service Two-Way Email JavaServer Pages.....	9-2
	Logging on to Self-Service Two-Way Email.....	9-3
	Creating a Self-Service Two-Way Email Template.....	9-4
10	Self-Service Two-Way Email Administrator Console	10-1
	Accessing the Administrator Console.....	10-2
	Using the Job Log.....	10-2
	Purging the Job Log.....	10-4
	Using the Event Log.....	10-5
	Purging the Event Log.....	10-6
	Checking the Status of a Job or Canceling a Job.....	10-7
A	ReportCaster API Debugging Techniques	A-1
	Tracing.....	A-2
	WebFOCUS Reporting Server Tracing.....	A-2
	IBITRACE Facility.....	A-2
	Servlet Tracing.....	A-4
	Distribution Server Tracing.....	A-4
	Distribution Server Tracing Files.....	A-4
	Trace Files for Individual Jobs.....	A-5
	Tracing Error Files.....	A-6
	Examining a Query String.....	A-6
	Using the API Query Tool.....	A-8
B	ReportCaster API Messages and Codes	B-1
	ReportCaster API Subroutine Messages.....	B-2
	DSTRUNER (DSTRUN Return Codes).....	B-3
	DSTMEMER (DSTMEM Return Codes).....	B-5
	DSTBLKER (DSTBULK Return Codes).....	B-5
	ReportCaster Bean API Messages.....	B-5
	ReportCaster Servlet API Messages.....	B-5
C	API-based MRE User	C-1
	Accessing API-based Schedules and Log Reports.....	C-2
	Maintaining a Server Schedule.....	C-3
	Server Schedule Options.....	C-4
	Viewing an API-based Log File.....	C-4
	Index	I-1

CHAPTER 1

Introducing the ReportCaster API

Topics:

- Terminology for the ReportCaster API
- Installation Requirements
- ReportCaster Bean API Overview
- ReportCaster Servlet API Overview
- ReportCaster API Subroutines Overview

The ReportCaster Application Programming Interface (API) provides ReportCaster capabilities to developers of self-service applications. The ReportCaster API enables developers to provide users with the ability to schedule and distribute reports over the Web.

The ReportCaster API is independent of the WebFOCUS Managed Reporting Environment (MRE) and Business Intelligence Dashboard (BID). Developers must provide a valid WebFOCUS logon to their ReportCaster API applications for ownership of schedules and Distribution Lists, and to implement security.

The ReportCaster API consists of ReportCaster Bean API, ReportCaster Servlet API, and ReportCaster API subroutines. Using Java Database Connectivity (JDBC), the ReportCaster API accesses a repository of tables that store scheduling and distribution information supplied by the application user. The ReportCaster API bundles the distribution information and sends it to the WebFOCUS Reporting Server to run.

Information Builders recommends that you use:

- ReportCaster Bean API for scheduling and log functions.
- ReportCaster Servlet API for Address Book functionality.

Terminology for the ReportCaster API

Throughout the rest of this manual, the following terminology is discussed:

- **Servlets**—programs that run on a Web server that modify a Web page before it is sent to the user who requested it.
- **Subroutines**—a subset of C-based API functions that call a servlet. Subroutines are called from a procedure (FOCEXEC) on the WebFOCUS Reporting Server.
- **JavaBeans**—an object-oriented programming interface from Sun Microsystems that enables you to build reusable applications.
- **Beans**—components (objects) written by a Java developer that extend JavaBeans components (for example, DSTSchedFactory).
- **Objects**—the building blocks of object-oriented programming, an object is a self-contained module of data and its associated processes.
- **Packages** —groups together related classes and subpackages.
- **Class**—a collection of objects that share the same characteristics.
- **Method**—a group of programming statements that perform a specific task (for example, getScheduleListHandler).
- **Properties**—attributes of a method that affect object processing (for example, IBIB_scheduleid).
- **JavaServer Pages (JSP)**—a technology for controlling the content and look of Web pages through the use of a servlet (for example, rcaster_list.jsp).
- **Document Type Definition (DTD)**—specifies the types of tags that can be included in your XML document, and how each tag is to be processed.

For more information about Java technology, visit <http://java.sun.com> on the Web.

Installation Requirements

The following are installation requirements for the ReportCaster API:

- Servlet-enabled Web server or application server.
- JDK 1.3.1 on your Web server or application server.

For more information, see the *WebFOCUS and ReportCaster Installation and Configuration* manual for your platform.

ReportCaster Bean API Overview

The ReportCaster API consists of JavaBeans components that are extended by Information Builders as Beans. The ReportCaster Bean API include all log and scheduling capabilities, including the ability to update and delete schedules, and query the ReportCaster Repository for information.

The ReportCaster Bean API enables developers to use JavaServer Pages (JSP) technology, a Java servlet, or a Java application to create an independent, customized application that offers the features of ReportCaster. The ReportCaster Bean API does not require a WebFOCUS cookie for user authentication. The DSTAuthenticate Bean includes methods that allow you to set the WebFOCUS Reporting Server user ID and password.

Scheduling Options

The scheduling options for the ReportCaster Bean API enable you to:

- Create a schedule.
- List the procedures (FOCEXECs) on the WebFOCUS Reporting Server available for scheduling.
- Specify whether to use a Distribution List, distribution file name, or single destination.
- Specify whether the report output should be sent within the body of the e-mail (inline) or as an attachment.
- Update a schedule.
- Delete a schedule.
- Retrieve a list of schedules.
- Retrieve the properties for a schedule.
- Retrieve the details of a schedule.
- Retrieve schedule data based on schedule field values or specific conditions.
- Run a job once at a specific time without storing the schedule information in the ReportCaster Repository.

Log File Options

The log file options for the WebFOCUS ReportCaster Bean API enable you to:

- Retrieve the list of schedules that have log information stored in the ReportCaster Repository.
- Retrieve log records and log messages for a specific process.
- Retrieve log records and log messages based on log field values and specific conditions.
- Delete a log for a specified process.
- Delete log records based on log field values and specific conditions.

For more information about the ReportCaster Bean API, see Chapter 2, *ReportCaster Bean API*.

ReportCaster Servlet API Overview

The ReportCaster Servlet API is a set of Java functions that provide ReportCaster capabilities to developers of self-service applications (such as customized applications driven by HTML and JavaScript).

The ReportCaster Servlet API provides developers with schedule, log, and Address Book functionality. However, Information Builders recommends that you use the ReportCaster Bean API for all scheduling and log functions.

For more information, see Chapter 3, *ReportCaster Servlet API*.

ReportCaster API Subroutines Overview

The ReportCaster API subroutines are a set of C-based API functions that call a servlet. The subroutines are called from a procedure (FOCEXEC) on the WebFOCUS Reporting Server, and they enable a user to:

- Create and maintain Distribution Lists that contain the addresses of recipients of scheduled reports. These may be public or private Distribution Lists.
- Immediately run and distribute a report.

For more information, see Chapter 4, *ReportCaster API Subroutines*.

CHAPTER 2

ReportCaster Bean API

Topics:

- Beans Within the `ibi.broker.beans` Package
- Beans Within the `ibi.broker.beans.handler` Package
- ReportCaster Repository Access
- Security for the ReportCaster Bean API
- Using the ReportCaster Bean API
- DTDs for ReportCaster XML Result Sets
- Processing ReportCaster Bean API Result Sets
- JavaServer Pages Technology

The ReportCaster Bean API consists of JavaBeans components. These JavaBeans components (or Beans) handle the application logic for a ReportCaster self-service application. For detailed online documentation about these Beans, developers can access:

- http://hostname/ibi_html/broker/docapibeans/index.html (where *hostname* is the host name of the Web server on which WebFOCUS is installed).
- *ReportCaster API for JavaBeans Components* on your documentation CD.

The ReportCaster Bean API is independent of the WebFOCUS Managed Reporting Environment (MRE) and Business Intelligence Dashboard (BID). It enables developers to use JSP technology, a Java servlet, or a Java application to create an independent, customized application that offers the features of ReportCaster.

Beans Within the ibi.broker.beans Package

The following table describes the Beans within the ibi.broker.beans package.

Bean Name	Description
DSTAPIStatus	Checks completion status for authentication and initialization. Detects exception messages and validates properties.
DSTAuthenticate	Provides basic service for security authentication and configuration information.
DSTBeanResult	Holds the results, condition codes, and messages for all functions in the ReportCaster Bean API.
DSTBeanHandler	Represents all handler objects that perform a function in the ReportCaster Bean API.
DSTLogFactory	Enables applications to set or get log properties and obtain any specified DSTBeanHandler to retrieve, delete, or update the BOTLOG repository table.
DSTSchedFactory	Enables applications to set or get schedule properties and obtain any specified DSTBeanHandler to retrieve, delete, or update the BOTSCHEM repository table.

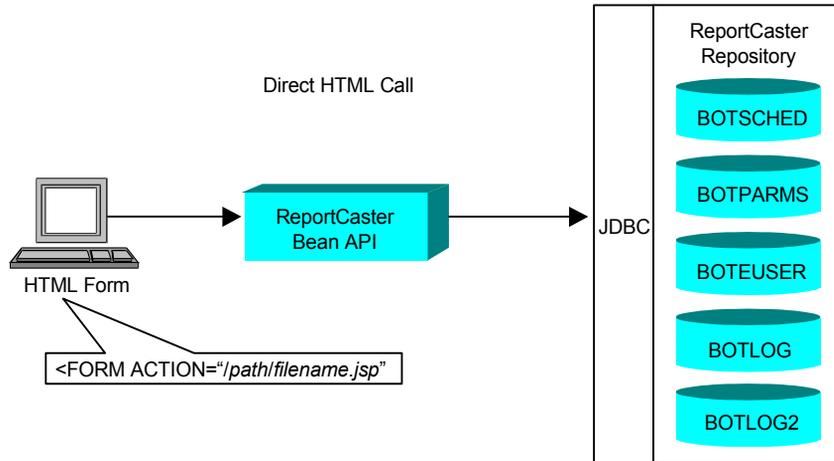
Beans Within the ibi.broker.beans.handler Package

The following table describes the Beans within the ibi.broker.beans.handler package.

Bean Name	Description
DSTPeriodSchedule	Is the handler used to obtain schedule or log information.
DSTSelectionHandler	Is the handler used to get the schedule list.

ReportCaster Repository Access

When the ReportCaster Repository is SQL-based, the ReportCaster Bean API uses JDBC to write to and read from the repository tables. The following diagram illustrates the communication by the ReportCaster Bean API to an SQL-based ReportCaster Repository.



When configuring ReportCaster with a FOCUS Repository, the ReportCaster Bean API connects to the WebFOCUS Reporting Server to communicate to the FOCUS Database Server (FDS) to write to and read from the FOCUS Repository data sources.

Reference ReportCaster Bean API Repository Tables

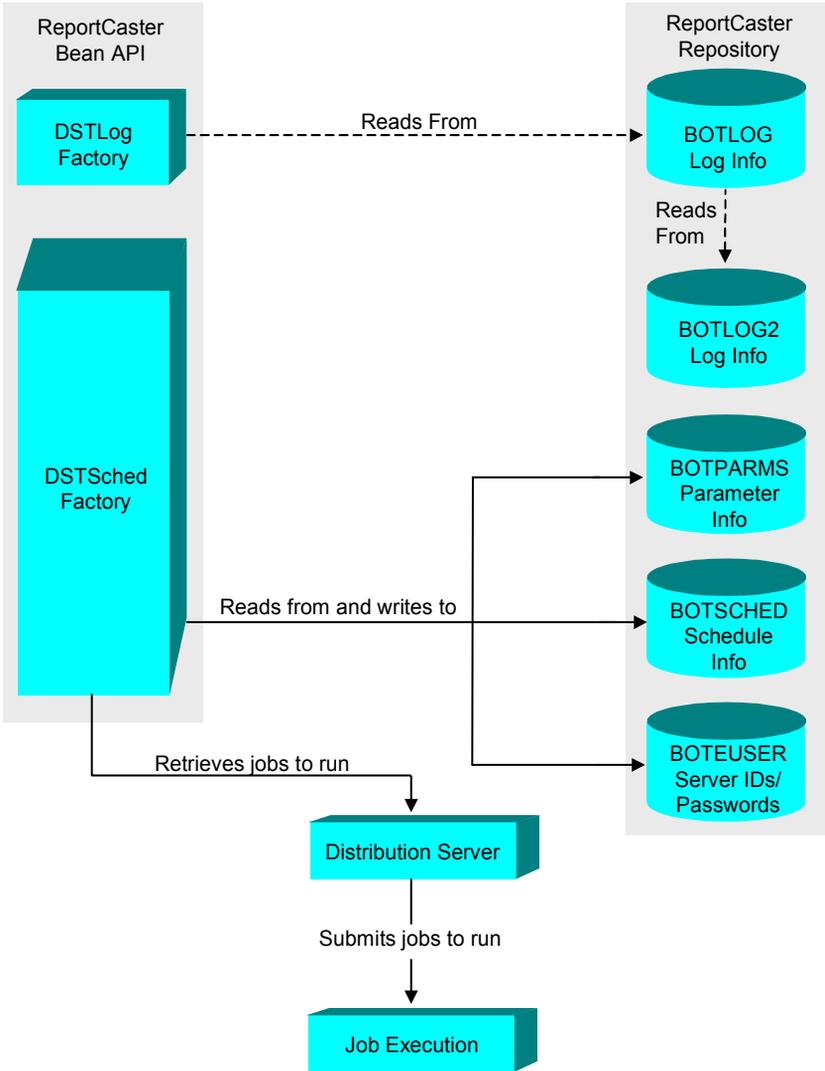
The following ReportCaster Repository tables are accessed by the ReportCaster Bean API:

Repository Table	Description	Bean Used
BOTSCHED	Contains scheduling information. Includes scheduling interval, procedures that are run before a report, and procedures that are run after a report.	DSTSchedFactory
BOTPARMS	Contains parameter information. Enables a user to schedule a report to run with certain parameter values.	DSTSchedFactory
BOTEUSER	Contains valid WebFOCUS Reporting Server user IDs and passwords.	DSTSchedFactory
BOTLOG	Contains log property information (job description, schedule ID, user IDs, and start and end time of the job).	DSTLogFactory
BOTLOG2	Contains information on the events related to the execution and distribution of a report. Note: The DSTLogFactory Bean has the ability to delete records in BOTLOG, which then would delete corresponding records in BOTLOG2.	DSTLogFactory

Reference

ReportCaster Bean API and Repository Tables

The following diagram illustrates the ReportCaster Bean API and the tables they access in the ReportCaster Repository.



Security for the ReportCaster Bean API

User authentication and configuration information must be established and validated before accessing the ReportCaster Bean API.

User Validation

The DSTAuthenticate Bean does not require a WebFOCUS cookie to validate user credentials on the WebFOCUS Reporting Server. The DSTAuthenticate Bean contains two methods, setUser and setPass, that allow you to set the WebFOCUS Reporting Server user ID and password within your JSP, Java servlet, or Java application. If you are coding a Java application, you must also use the setAgentProperty method.

If you are coding a JSP or servlet, you may use a WebFOCUS cookie to store the user's WebFOCUS Reporting Server user ID and password. When a WebFOCUS cookie is established, you must use the setInitParameter and setRequest methods in order to use the WebFOCUS cookie values.

A WebFOCUS cookie is established when a valid WebFOCUS logon occurs using the WF_SIGNON action. A sample logon form, rbalogon.htm, is distributed with ReportCaster in *drive:\ibi\WebFOCUS\rel\ibi_html\broker*

where:

drive

Is the letter of the drive on which WebFOCUS is installed.

rel

Is the WebFOCUS release.

Configuration Information

The configuration information needed by the DSTAuthenticate Bean is the value for the ReportCaster Distribution Server host name and port (AGENT_NODE and AGENT_PORT). These values allow the ReportCaster Bean API to connect to the Distribution Server to request the configuration information specified in the ReportCaster configuration file. This file is located in:

drive:\ibi\distributionserver\rel\cfg\bkrsched.cfg

where:

drive

Is the letter of the drive on which the Distribution Server is installed.

rel

Is the WebFOCUS release.

This configuration information is needed to communicate to the ReportCaster Repository and to connect to the WebFOCUS Reporting Server.

DSTAuthenticate Bean Properties

The DSTAuthenticate Bean has two methods that are used to obtain configuration information:

- **setInitParameter**—uses the location of the WebFOCUS Client. The location of the WebFOCUS Client enables the DSTAuthenticate Bean to locate the WebFOCUS Client configuration file `ibidir.wfs`, which contains the parameters and values for `AGENT_NODE` and `AGENT_PORT` when WebFOCUS is installed with ReportCaster.

The following is an example of the section of JSP code that performs authentication. The WebFOCUS Client is installed on the `c:` drive.

```
<jsp:useBean id="auth" scope="session"
class="ibi.broker.beans.DSTAuthenticate" />

<jsp:useBean id="factory" scope="session"
class="ibi.broker.beans.DSTSchedFactory" />

<%
    factory.setAuthenticate(auth);
    factory.clear();
    auth.setRequest(request);
%>

<jsp:setProperty name = "auth" property="initParameter"
value="c:/ibi/client51/conf/web/cgi" />
```

- **setAgentProperty**—specifies the host name and port of the Distribution Server. The parameter value must be specified as `hostname:port`. This method must be used when coding a Java application.

The following is an example of the section of JSP code that performs authentication. The default values of `localhost` and `8200` are used for the Distribution Server host name and installation port.

```
<jsp:useBean id="auth" scope="session"
class="ibi.broker.beans.DSTAuthenticate" />

<jsp:useBean id="factory" scope="session"
class="ibi.broker.beans.DSTSchedFactory" />

<%
    factory.setAuthenticate(auth);
    factory.clear();
    auth.setRequest(request);
%>

<jsp:setProperty name = "auth" property="agentProperty"
value="localhost:8200" />
```

Using the ReportCaster Bean API

JavaServer Pages (JSP) are used by the ReportCaster Bean API to return result sets to the user. You cannot directly access the JavaBeans components from a directory. They reside in the DSTServlets.jar file, located by default in the WebFOCUS javaassist directory, and are unpackaged at run-time. For more information about the sample ReportCaster Bean API application, see Chapter 5, *ReportCaster API JSP Samples*.

Procedure

How to Use the ReportCaster Bean API

To use the ReportCaster Bean API within your JSP, Java servlet, or Java application, follow the steps below. For illustration purposes, sections of code from a sample JSP follow each step.

1. Import Bean packages to make them available to your Java program (JSP, Java servlet, or Java application).

```
<%@ page language="java" import="ibi.broker.beans.*" %>
```

2. Create a DSTAuthenticate object that contains all the property information needed for security authentication to the Distribution Server.

```
<jsp:useBean id="auth"  
scope="session" class="ibi.broker.beans.DSTAuthenticate" />
```

Note: If you are using a Java application, you must use the setUser and setPass methods for user authentication, and the setAgentProperty method for configuration. All other steps in this procedure are the same regardless of whether you are using a JSP, a Java servlet, or a Java application.

3. Create an object for the Bean from which you want to perform functions (scheduling and log). For example, in the following code, the factory object contains all the property information needed for scheduling.

```
<jsp:useBean id="factory" scope="session"  
class="ibi.broker.beans.DSTSchedFactory" />
```

4. Submit an authentication object into the object created in step 3.

If you are using the Common Graphical Interface (CGI):

```
<%  
    factory.setAuthenticate(auth);  
    factory.clear();  
    auth.setRequest(request);  
%>
```

If you are not using the CGI:

```
factory.setAuthenticate(auth);
```

5. Specify connection and configuration information.

If you are using the Common Graphical Interface (CGI):

Log on to the WebFOCUS Reporting Server using IBIC_user and IBIC_pass in a logon page such as rbalogon.htm or rbaindex.htm (for more information, see Chapter 5, *ReportCaster API JSP Samples*). A successful logon creates a WebFOCUS cookie (WF_COOKIE).

```
<jsp:setProperty name = "auth" property="initParameter"
value="drive:/ibi/client51/conf/web/cgi" />
```

where:

drive

Is the letter of the drive on which the WebFOCUS Client is installed.

If you are not using the CGI:

Create a logon page that collects the node, port, user, and password.

For example, using Java:

```
<%
String Node = request.getParameter("IBIB_node");
// Agent_node for distribution
String Port = request.getParameter("IBIB_port");
// Agent_port for distribution
String User = request.getParameter("IBIB_user");
// WebFOCUS Reporting Server ID
String Pass = request.getParameter("IBIB_pass");
// WebFOCUS Reporting Server password
auth.setAgentProperties(Node:"Port");
auth.setUser(User);
auth.setPass(Pass);
%>
```

6. For each function (or method) you want to run, perform the following:

- a. Using the object created in step 3, set the properties for the function you want to perform. Following is an example of using the values on an HTML form to set the properties for the factory object.

```
<jsp:setProperty name = "factory" property="*" />
```

- b. Create a request object in the DSTBeanHandler that will contain the function specified in the factory object.

```
DSTBeanHandler app = factory.createScheduleHandler();
```

- c. In the DSTBeanHandler object, issue the processRequest method to run the Bean function specified in the factory object. For example, to create a new schedule, process the request object in Step b as follows:

```
app.processRequest();
```

- d. Issue the `getAPIStatus` method of the `DSTBeanHandler` object to obtain the `DSTAPIStatus` Bean. This describes the status of your request.

```
DSTAPIStatus status = app.getAPIStatus();
int code = status.getErrorCode();
```

or

```
String message = status.getErrorMessage();
```

- e. For functions that return data, such as log functions, you must first retrieve the error code of the `DSTAPIStatus` Bean to obtain the return code. Next, compare the return code to the variable `DSTAPIStatus.NO_ERROR`. There are two possible outcomes:
- The return code and the `DSTAPIStatus.NO_ERROR` variable are equal, meaning that your request was successful and you can now obtain your answer set.
 - The return code and `DSTAPIStatus` are not equal, meaning that your request failed. Use the `getErrorMessage` method in the `DSTAPIStatus` object to obtain a detailed message.

The following section of code illustrates the error recovery process.

```
int code = status.getErrorCode();
if (code == DSTAPIStatus.NO_ERROR) {
    String back = request.getParameter("CASTER_backlink2");
%>
    <jsp:forward page="<%= back %%" />
<%
}
else {
%>
    <jsp:forward page="rcaster_output.jsp">
        <jsp:param name="CASTER_errormessage" value="<%=
status.getErrorMessage() %%" />
    </jsp:forward>
<%
}
%>
```

Note: Messages implemented for `DSTAPIStatus` object are not currently internationalized.

- f. To obtain an answer set, issue the `getBeanResult` of the `DSTBeanHandler` object. This will return a `DSTBeanResult` object. The result set can be obtained in XML, as a `ReportCaster` result set, or sent to the browser in HTML format
 - To obtain your result as XML, use the `getXML` method of the `DSTBeanResult`. For more information about the DTD for the XML result sets returned by the `ReportCaster` Bean API, see *DTDs for ReportCaster XML Result Sets* on page 2-12.
 - To obtain your result as a `ReportCaster` result set, see *Processing ReportCaster Bean API Result Sets* on page 2-18.
 - To send your result to the browser, code a JSP. HTML code incorporated in the JSP is used to display the result set. For more information about the methods and properties that may be accessed in your JSP, see http://hostname/ibi_html/broker/docapibeans/index.html (where *hostname* is the host name of the Web server on which WebFOCUS is installed), *ReportCaster API for JavaBeans Components* on your documentation CD, and Chapter 5, *ReportCaster API JSP Samples*.

DTDs for ReportCaster XML Result Sets

The following section provides the Document Type Definitions (DTDs) for the XML result sets returned by the ReportCaster Bean API. Some of the DTD elements listed may have fixed values. These are noted by comments within the DTDs. You can also obtain the appropriate values by checking the 'get' and 'set methods' for the property in either DSTSchedFactory or DSTLogFactory.

Note: Elements that have a description that starts with 'Interval value: are in development and are not supported for customer use in WebFOCUS Version 4 Release 3.6.

Reference

DTD for Schedule List

```
<?xml version="1.0" encoding="UTF-8"?>
<!--DTD generated by XML Spy v4.0.1 U (http://www.xmlspy.com)-->
<!ELEMENT IBIB_active (#PCDATA)> ('Flag indicating the status of the job
as active or inactive')
<!ELEMENT IBIB_jobdesc (#PCDATA)> ('Description of job being scheduled')
<!ELEMENT IBIB_jobname (#PCDATA)> ('Name of procedure, focexec or
filename, being scheduled')
<!ELEMENT IBIB_method (#PCDATA)> ('Distribution method: FTP, PRINT or
MAIL')
<!ELEMENT IBIB_mreuser (#PCDATA)> ('Internal value: used for MRE user ID
to establish ownership of job')
<!ELEMENT IBIB_scheduleid (#PCDATA)> ('API-generated key that identifies
a scheduled job')
<!ELEMENT IBIB_userid (#PCDATA)> ('WebFOCUS Reporting Server user ID')
<!ELEMENT reportcaster (schedule+)>
<!ELEMENT schedule (IBIB_scheduleid, IBIB_jobname, IBIB_jobdesc,
IBIB_active, IBIB_method, IBIB_userid, IBIB_mreuser)>
```

Reference **DTD for Schedule Properties**

```

<?xml version="1.0" encoding="UTF-8"?>
<!--DTD generated by XML Spy v4.0.1 U (http://www.xmlspy.com)-->
<!ELEMENT IBIB_active (#PCDATA)> ('Flag indicating the status of the job
as active or inactive')
<!ELEMENT IBIB_byfield (#PCDATA)> ('Indicates whether the report will be
burst or not')
<!ELEMENT IBIB_distlist (#PCDATA)> ('Distribution list containing
recipients of the report')
<!ELEMENT IBIB_fexorigin (#PCDATA)> ('Indicates whether procedure
scheduled from Server, MRE or Desktop. API currently supports only
Server')
<!ELEMENT IBIB_interval (#PCDATA)> ('Period of time:that execution and
distrib are based:Hourly, Daily, Weekly, Monthly, Yearly, Once')
<!ELEMENT IBIB_jobdesc (#PCDATA)> ('Description of job being scheduled')
<!ELEMENT IBIB_jobname (#PCDATA)> ('Name procedure:FOCEXEC:being
scheduled')
<!ELEMENT IBIB_listtype (#PCDATA)> ('Indicates whether report distributed
by mail is sent inline or as an attachment')
<!ELEMENT IBIB_method (#PCDATA)> ('Distribution method: FTP, PRINT or
MAIL')
<!ELEMENT IBIB_mreuser (#PCDATA)> ('Internal value: MRE user ID
indicating owner of job')
<!ELEMENT IBIB_nextruntime (#PCDATA)> ('Indicates the next time the job
is scheduled to run. Stored in GMT')
<!ELEMENT IBIB_priority (#PCDATA)> ('Priority level of job scheduled to
run')
<!ELEMENT IBIB_scheduleid (#PCDATA)> ('API-generated key that identifies
a scheduled job')
<!ELEMENT IBIB_sendformat (#PCDATA)> ('Format of the distribution report
output')
<!ELEMENT IBIB_userid (#PCDATA)> ('WebFOCUS Reporting Server user ID')
<!ELEMENT reportcaster (schedule)>
<!ELEMENT schedule (IBIB_scheduleid, IBIB_jobname, IBIB_jobdesc,
IBIB_active, IBIB_userid, IBIB_nextruntime, IBIB_fexorigin, IBIB_method,
IBIB_listtype, IBIB_distlist, IBIB_priority, IBIB_byfield,
IBIB_sendformat, IBIB_interval, IBIB_mreuser)>

```

Reference **DTD for Schedule Detail**

```
<?xml version="1.0" encoding="UTF-8"?>
<!--DTD generated by XML Spy v4.0.1 U (http://www.xmlspy.com)-->
<!ELEMENT IBIB_active (#PCDATA)> ('Flag indicating the status of the
job')
<!ELEMENT IBIB_alertjob (#PCDATA)> ('Internal value: Flag indicating
whether the schedule is for an Alert')
<!ELEMENT IBIB_asvalue EMPTY> ('Name of FTP index file for a bursted
report')
<!ELEMENT IBIB_byfield (#PCDATA)> ('Indicates whether the report will be
burst or not')
<!ELEMENT IBIB_dates (#PCDATA)> ('Day of the month the report request
will run')
<!ELEMENT IBIB_deactivate (#PCDATA)> ('Flag indicating status of the job
is active or inactive')
<!ELEMENT IBIB_distlist (#PCDATA)> ('Distribution list containing
recipients of the report')
<!ELEMENT IBIB_domainhref EMPTY> ('Internal value: Location of domain in
Managed Reporting Repository')
<!ELEMENT IBIB_enddate (#PCDATA)> ('Date when the job stops running.
Stored in GMT')
<!ELEMENT IBIB_endtime (#PCDATA)> ('Time when the job stops running.
Stored in GMT')
<!ELEMENT IBIB_fexorigin (#PCDATA)> ('Indicates where
procedure:FOCEXEC:is located:Server,MRE, or Desktop. Currently only
supports Server')
<!ELEMENT IBIB_folderhref EMPTY> ('Internal value: Location of report
within Managed Reporting domain')
<!ELEMENT IBIB_frequency (#PCDATA)> ('Number of time the report is
executed and distributed within schedule interval:IBIB_interval')
<!ELEMENT IBIB_ftphost (#PCDATA)> ('Name of FTP server that distributes
the report')
<!ELEMENT IBIB_ftplocation (#PCDATA)> ('Destination of FTP-Distributed
report')
<!ELEMENT IBIB_ftpPASS (#PCDATA)> ('Password associated with FTP user
ID')
<!ELEMENT IBIB_ftpPublish (#PCDATA)> ('Indicates whether to distribute
FTP as Prepared Report to MRE')
<!ELEMENT IBIB_ftpPublishName EMPTY> ('Name of Prepared Report displayed
in MRE')
<!ELEMENT IBIB_ftpuser (#PCDATA)> ('User ID authorized for FTP transfer')
<!ELEMENT IBIB_interval (#PCDATA)> ('Period of time:that execution and
distrib are based:Daily, Weekly, Hourly, Monthly, Once')
<!ELEMENT IBIB_jobdesc (#PCDATA)> ('Description of job being scheduled')
<!ELEMENT IBIB_jobname (#PCDATA)> ('Name of procedure:FOCEXEC:being
scheduled')
```

```

<!ELEMENT IBIB_listtype (#PCDATA)> ('Indicates whether report distributed
by mail is sent inline or as an attachment')
<!ELEMENT IBIB_mailcompany EMPTY> ('The company that the sender is
associated with')
<!ELEMENT IBIB_mailfrom EMPTY> ('Email address for returned mail')
<!ELEMENT IBIB_mailhost (#PCDATA)> ('Name of mail server that distributes
the report')
<!ELEMENT IBIB_mailssubject EMPTY> ('Text that describes content or
purpose of email')
<!ELEMENT IBIB_method (#PCDATA)> ('Distribution method: FTP, PRINT or
MAIL')
<!ELEMENT IBIB_mrenode EMPTY> ('Internal value: MRE only:Node and port
for web server where MRE installed')
<!ELEMENT IBIB_mreuser (#PCDATA)> ('Internal value: MRE only:MRE user ID
indicating owner of job')
<!ELEMENT IBIB_mroutputfolder EMPTY> ('Internal value: MRE only:Managed
Reporting domain folder for scheduled procedure:FOCEXEC')
<!ELEMENT IBIB_multipart (#PCDATA)> ('Indicates whether the report should be
in the body of the mail or as an attachment')
<!ELEMENT IBIB_nextruntime (#PCDATA)> ('Indicates the next time the job
is scheduled to run')
<!ELEMENT IBIB_notifyaddress EMPTY> ('E-mail address of person who will
recieve notification')
<!ELEMENT IBIB_notifybrief EMPTY> ('E-mail address of device. Cellphone,
pager, or handheld that will recieve the brief notice')
<!ELEMENT IBIB_notifyflag (#PCDATA)> ('Flag that controls notification of
job status:Send notification Never,Always,On Error')
<!ELEMENT IBIB_notifyreply EMPTY> ('Email address for return mail')
<!ELEMENT IBIB_notifysubject EMPTY> ('Text that describes content or
purpose of notice')
<!ELEMENT IBIB_paramname (#PCDATA)> ('Name of parameter passed to
Distribution Server')
<!ELEMENT IBIB_paramvalue (#PCDATA)> ('Value of parameter passed to
Distribution Server')
<!ELEMENT IBIB_postrpc1 EMPTY> ('Name of procedure that will run after
scheduled job')
<!ELEMENT IBIB_postrpc2 EMPTY> ('Name of procedure that will run after
scheduled job')
<!ELEMENT IBIB_prerpc1 EMPTY> ('Name of procedure that will run prior to
scheduled job')
<!ELEMENT IBIB_prerpc2 EMPTY> ('Name of procedure that will run prior to
scheduled job')
<!ELEMENT IBIB_priority (#PCDATA)> ('Priority level of job scheduled to
run')
<!ELEMENT IBIB_recovery (#PCDATA)> ('Indicates whether to recover the
job')
<!ELEMENT IBIB_scheduleid (#PCDATA)> ('API-generated key that identifies
a scheduled job')
<!ELEMENT IBIB_sendformat (#PCDATA)> ('Format of the distribution report
output')
<!ELEMENT IBIB_startdate (#PCDATA)> ('Date when the schedule started
running')

```

```
<!ELEMENT IBIB_starttime (#PCDATA)> ('Time when the schedule started
running')
<!ELEMENT IBIB_sync (#PCDATA)> ('Internal value: Indicates whether PDA
sync will be used')
<!ELEMENT IBIB_updatedate (#PCDATA)> ('Indicates when the schedule was
last updated')
<!ELEMENT IBIB_updatetime (#PCDATA)> ('Indicates the time the schedule was
last updated')
<!ELEMENT IBIB_userid (#PCDATA)> ('WebFOCUS Reporting Server user ID')
<!ELEMENT IBIB_weekdays (#PCDATA)> ('Days of the week the report request
will run')
<!ELEMENT parameterelement (IBIB_paramname, IBIB_paramvalue)>
<!ELEMENT parameters (parameterelement+)>
<!ELEMENT reportcaster (schedule)>
<!ELEMENT schedule (IBIB_scheduleid, IBIB_jobname, IBIB_userid,
IBIB_mreuser, IBIB_jobdesc, IBIB_listtype, IBIB_updatedate,
IBIB_updatetime, IBIB_fexorigin, IBIB_folderhref, IBIB_domainhref,
IBIB_mrenode, IBIB_frequency, IBIB_interval, IBIB_weekdays, IBIB_dates,
IBIB_startdate, IBIB_starttime, IBIB_enddate, IBIB_endtime,
IBIB_nextruntime, IBIB_active, IBIB_priority, IBIB_recovery, IBIB_prerpcl,
IBIB_prerpcl2, IBIB_postrpcl, IBIB_postrpcl2, IBIB_distlist, IBIB_byfield,
IBIB_asvalue, IBIB_sendformat, IBIB_method, IBIB_notifyaddress,
IBIB_notifybrief, IBIB_notifyreply, IBIB_notifysubject, IBIB_notifyflag,
IBIB_mroutputfolder, IBIB_mailhost, IBIB_mailfrom, IBIB_mailsubject,
IBIB_mailcompany, IBIB_ftphost, IBIB_ftplocation, IBIB_ftpuser,
IBIB_ftppass, IBIB_ftppublish, IBIB_ftppublishname, IBIB_deactivate,
IBIB_sync, IBIB_multipart, IBIB_alertjob, parameters)>
```

Reference

DTD for ReportCaster Log List

```
<?xml version="1.0" encoding="UTF-8"?>
<!--DTD generated by XML Spy v4.0.1 U (http://www.xmlspy.com)-->
<!ELEMENT IBIB_jobdesc (#PCDATA)> ('Description of job being logged')
<!ELEMENT IBIB_jobname (#PCDATA)> ('Name of procedure:FOCEXEC that ran')
<!ELEMENT IBIB_mreuser (#PCDATA)> ('Internal value: MRE user ID
indicating owner of job:Internal value assigned to API schedules')
<!ELEMENT IBIB_scheduleid (#PCDATA)> ('API-generated key that identifies a
scheduled job')
<!ELEMENT IBIB_userid (#PCDATA)> ('WebFOCUS Reporting Server user ID')
<!ELEMENT log (IBIB_scheduleid, IBIB_jobname, IBIB_jobdesc, IBIB_userid,
IBIB_mreuser)>
<!ELEMENT reportcaster (log+)>
```

Reference**DTD for ReportCaster Log Process List**

```
<?xml version="1.0" encoding="UTF-8"?>
<!--DTD generated by XML Spy v4.0.1 U (http://www.xmlspy.com)-->
<!ELEMENT IBIB_endstamp (#PCDATA)> ('Date and time the job finished
running')
<!ELEMENT IBIB_error (#PCDATA)> ('Indicates if a fatal error has occurred
for this job')
<!ELEMENT IBIB_jobnumber (#PCDATA)> ('Unique 12 digit schedule ID for the
job')
<!ELEMENT IBIB_startstamp (#PCDATA)> ('Date and time the job started
running')
<!ELEMENT IBIB_twoway (#PCDATA)> ('Indicates whether the job was a TwoWay
Email job')
<!ELEMENT log (IBIB_jobnumber, IBIB_startstamp, IBIB_endstamp,
IBIB_twoway, IBIB_error)>
<!ELEMENT reportcaster (log+)>
```

Reference**DTD for ReportCaster Process Log**

```
<?xml version="1.0" encoding="UTF-8"?>
<!--DTD generated by XML Spy v4.0.1 U (http://www.xmlspy.com)-->
<!ELEMENT IBIB_counter (#PCDATA)> ('Internal use: unique key created by
ReportCaster')
<!ELEMENT IBIB_endstamp (#PCDATA)> ('Date and time the job finished
running')
<!ELEMENT IBIB_error (#PCDATA)> ('Indicates if a fatal error has occurred
for this job')
<!ELEMENT IBIB_jobdesc (#PCDATA)> ('Description of job being logged')
<!ELEMENT IBIB_jobname (#PCDATA)> ('Name of FOCEXEC procechure being
logged')
<!ELEMENT IBIB_jobnumber (#PCDATA)> ('Unique 12 digit schedule ID for the
job')
<!ELEMENT IBIB_message (#PCDATA)> ('Text returned by API indicating a
successful completion')
<!ELEMENT IBIB_mreuser (#PCDATA)> ('MRE user ID indicating owner of
job:Internal value assigned to API schedules')
<!ELEMENT IBIB_msgcode (#PCDATA)> ('Message code for this log message
record')
<!ELEMENT IBIB_msgerror (#PCDATA)> ('Error Message')
<!ELEMENT IBIB_scheduleid (#PCDATA)> ('API-generated key that identifies
a scheduled job')
<!ELEMENT IBIB_startstamp (#PCDATA)> ('Date and time the job started
running')
<!ELEMENT IBIB_twoway (#PCDATA)> ('Internal value: Indicates whether the
job was a TwoWay Email job')
<!ELEMENT IBIB_userid (#PCDATA)> ('WebFOCUS Reporting Server user ID')
<!ELEMENT log (IBIB_jobnumber, IBIB_jobname, IBIB_jobdesc,
IBIB_scheduleid, IBIB_userid, IBIB_mreuser, IBIB_startstamp,
IBIB_endstamp, IBIB_twoway, IBIB_error, logdata)>
<!ELEMENT logdata (logdataelement+)>
<!ELEMENT logdataelement (IBIB_counter, IBIB_msgcode, IBIB_message,
IBIB_msgerror)>
<!ELEMENT reportcaster (log)>
```

Processing ReportCaster Bean API Result Sets

ReportCaster Bean API result sets are organized by group. Each group signifies a certain class of information, such as scheduling or log information. ReportCaster Bean API result sets may consist of one or more groups of information. For example, getting the complete set of data for a given schedule may consist of two groups of information: one consisting of a single row of data describing the schedule, and another consisting of 0 or more rows for each parameter the schedule contains. Currently, there are four possible groups, which contain information about:

1. **Schedule**—Basic scheduling.
2. **Parameter**—Schedule parameters.
3. **Log**—Each process that is logged.
4. **LogMessage**—Detail records for each process that is logged.

The ReportCaster Bean API developer only needs to be concerned about the group when retrieving the number of rows of information in a result set. Since a ReportCaster Bean API result set may contain more than one group of information, it is necessary to tell the API which group of information you want the number of rows for. Use the `getRecordNumber(int id)` method of the `DSTBeanResult` class. This method returns the number of rows in the group of information specified by the `id` parameter. Use the constants in the `DSTSchedFactory` class and the `DSTLogFactory` class to specify the group ID.

In the `DSTSchedFactory` class, use either the `Schedule` integer constant or the `Parameter` integer constant. When you want to know the number of rows of schedule information, use the `Schedule` constant, and when you want to know the number of parameters, specify the `Parameter` integer constant.

In the `DSTLogFactory` class, use either the `Log` integer constant or the `LogMessage` integer constant. When you want to know the number of rows of log information, use the `Log` constant and when you want to know the number of log message records, specify the `LogMessage` constant.

For example, to retrieve the number of rows of parameter information for a `DSTBeanResult` object named `result`, code the following:

```
int numRows = result.getRecordNumber(DSTSchedFactory.Parameter);
```

Retrieving Column Data

In order to retrieve column data from ReportCaster Bean API result sets using the `DSTBeanResult` object, you need the:

- Number of rows of data.
- Names of the fields you want to retrieve from the result set.

Obtaining the Number of Rows of Data

When obtaining the number of rows of data, the result set can consist of a single group or two groups:

Single group:

To obtain the number of rows of data, use the `getRecordNumber()` method of the `DSTBeanResult` class. This method takes no parameters and returns an `int` value of the number of rows. You can use this method with `DSTSchedFactory` when you have obtained a `DSTBeanHandler` using either the `getScheduleList` method or the `getScheduleProperty` method. You can also use this method with `DSTLogFactory` if you have obtained a `DSTBeanHandler` using the `getLogListHandler` method or the `getProcessLogHandler` method.

Two groups:

There are two situations when you cannot determine the number of rows using the simple form of the `getRecordNumber` method as outlined above. The first is when you are retrieving all of the schedule data for a given schedule using the `getScheduleDataHandler` method of the `DSTSchedFactory` class. The second is when you are using the `getProcessLogHandler` method of the `DSTLogFactory` class. In these two cases, there are two sets of data in the result set and there are two groups of information that must be retrieved.

When you use the `getScheduleDataHandler` method of the `DSTSchedFactory` class, there is one group of information for the schedule data returned and another group of data for the parameters for the schedule. In this case, the number of rows in the result set for the schedule information is always one. In order to find out the number of parameters for the schedule, you need to use the `getRecordNumber(int id)` method of the `DSTBeanResult` class. This method takes an `int` as a parameter and returns an `int` containing the number of rows. The `int` parameter is the group ID and is specified using static integer constants in the `DSTSchedFactory` class. To get the number of rows of schedule information, use the `Schedule` `int`. To obtain the number of rows of parameter information, use the `Parameter` `int`.

Example

Retrieving the Number of Rows With Parameter Information

If there is a `DSTBeanResult` variable named `result` and you want to know the number of rows with parameter information, code the following:

```
int numParmRows = result.getRecordNumber(DSTSchedFactory.Parameter);
```

When you use the `getLogProcessHandler` method of the `DSTLogFactory` class, there is one group of information for the Log data returned and another group of data for the Log message records for the process. In this case, the number of rows in the result set for the Log data information is always one. In order to find out the number of log message records for the process, use the `getRecordNumber(int id)` method of the `DSTBeanResult` class. In this case, the `int` parameter is specified using static ints in the `DSTLogFactory` class. To get the number of rows of log data information, use the `Log` `int`. To obtain the number of rows of log message information, use the `LogMessage` `int`.

Example **Retrieving the Number of Rows of Log Message Information**

If there is a DSTBeanResult variable named result and you want to know the number of rows of Log message information, code the following:

```
int numLogMessages = result.getRecordNumber(DSTLogFactory.LogMessage);
```

Using Property Names to Retrieve Column Data

All column data returned by the ReportCaster Bean API is returned as Strings.

There are two methods you can use to retrieve column data from a DSTBeanResult object containing an answer set.

- Use the getString(int index, String apiname) method of the DSTBeanResult class to retrieve a single column value from a single row. Retrieve values by specifying the row number and the column name. The row numbers begin with 0.

For example, if you wanted to retrieve the job description from a DSTBeanResult object after using the getScheduleListHandler method of the DSTschedFactory class, you would code the following (assuming a DSTBeanResult variable named result):

```
String jobDescription = result.getString(3, "IBIB_jobdesc");
```

This statement returns the job description from row 3 of the schedule data.

- Use the getString(String apiname) to return a String array of all values for a given column in the entire result set. The array of values is ordered the same way as the result set. The 0 element in the result set is the 0 element of the array. The fifth element of the result set is the fifth element of the array. For example, to retrieve an array of all job names from a DSTBeanResult object after using the getScheduleListHandler method of the DSTSchedFactory class, code the following (assuming a DSTBeanResult variable named result):

```
String[] jobNames = getString("IBIB_jobname");
```

This statement returns an array of all job names in the result set.

ReportCaster API JSP samples (for example, rcaster_detail.jsp, rcaster_logplist.jsp, rcaster_loglist.jsp, and rcaster_property.jsp) show in detail how to examine ReportCaster result sets. For more information, see Chapter 5, *ReportCaster API JSP Samples*.

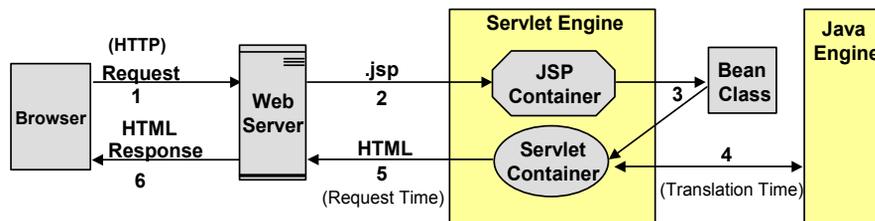
JavaServer Pages Technology

JavaServer Pages (JSP) are most commonly used for interactive user interfaces accessible from a Web server. Using JSP, application developers can separate the presentation logic from the processing logic. The presentation logic is the user interface, which can be a combination of HTML and JavaScript coded in the JSP. The processing logic is internal to the ReportCaster Bean API. For details about the required parameters for the JSP samples, see Chapter 5, *ReportCaster API JSP Samples*.

Execute a JSP by calling it directly from an HTML form or from the browser's URL address. JSP provides the following advantages:

- **Performance.** JSP are compiled dynamically into a servlet on the Web server's servlet engine the first time it is accessed. It then remains in memory for all users until the Web server or servlet engine is recycled.
- **Availability.** Define the scope of the request to be maintained for the entire browser session so that authentication is performed only once for a series of JSP.
- **Portability.** A JSP may be developed on one platform and deployed to another.

The following diagram illustrates the processing that takes place using JSP technology:



1. A JSP is invoked through a URL in the browser. The page is sent to the Web server using HTTP.
2. The Web server detects the file type of .jsp and sends the JSP to the Servlet engine (for example, JDK 1.3.1).

Note: New Atlanta ServletExec is used as a plug-in to the IIS Web server in order to communicate to the Servlet engine. ServletExec includes a JSP container and a servlet container.
3. The JSP accesses the Bean class and passes the information to the servlet container.
4. The servlet is then compiled by the Java engine and loaded into the servlet container. This process is known as the translation time.
5. The servlet runs and generates HTML, which is sent back to the Web server. This process is known as the request time.
6. The Web server sends the HTML back to the browser.

Calling a JSP From an HTML Form or URL

Call a JSP by:

- Specifying an absolute address for the Web server.
- Specifying a relative address for the Web server.
- Calling it directly from your browser's URL.

For details on specific JSP samples, see Chapter 5, *ReportCaster API JSP Samples*.

Syntax

How to Call a JSP From an HTML Form

Specifying an Absolute Address:

```
<FORM ACTION="http://hostname/path/filename.jsp"
```

where:

hostname

Is the host name of the Web server.

path

Is a location that is accessible to the Web server.

filename.jsp

Is the name of your JSP.

Specifying a Relative Address:

```
<FORM ACTION="/path/filename.jsp"
```

where:

path

Is a location that is accessible to the Web server.

filename.jsp

Is the name of your JSP.

Syntax

How to Call a JSP From a URL

```
http://hostname/path/filename.jsp
```

where:

hostname

Is the host name of the Web server.

path

Is a location that is accessible to the Web server.

filename.jsp

Is the name of your JSP.

Using a JSP to Set Bean Properties

Using a JSP, your HTML calling form can set, get, or prompt users to set the values of the properties to pass to the ReportCaster Bean API. Some parameter values are required.

- If you do not supply a required value but a default exists, the Bean will use the default.
- If you do not supply a required value and no default exists, the Bean will return a message.

For more information about setting Bean properties, see http://hostname/ibi_html/broker/docapibean/index.html (where *hostname* is the host name of the Web server on which WebFOCUS is installed), *ReportCaster API for JavaBeans Components* on your documentation CD, and Chapter 5, *ReportCaster API JSP Samples*.

CHAPTER 3

ReportCaster Servlet API

Topics:

- ReportCaster Servlet API Descriptions
- ReportCaster Repository Tables Accessed by the Servlets
- Servlet Security
- Calling a Servlet From an HTML Form
- Maintaining and Displaying a Distribution List With a Servlet
- Maintaining Distribution Lists Using the DSTDLBULK Servlet
- Maintaining Single Distribution List Members Using the DSTDLMEM Servlet
- Displaying a Distribution List Using the DSTDLLIST Servlet
- Scheduling a Job Using the DSTSCHED Servlet
- Setting the Status of a Job Using the DSTACTIVE Servlet
- Immediately Running a Scheduled Job Using the DSTRUNNOW Servlet
- Displaying Log Information Using the DSTLOG Servlet

The ReportCaster Servlet API enables application users to:

- Create, maintain, and display Distribution Lists that contain the addresses of recipients of scheduled reports. These may be public or private lists.
- Schedule the time and frequency of report execution and distribution. These reports may be scheduled immediately or as deferred reports.
- View information about the execution and distribution of reports.

Note: The ReportCaster Servlet API is functionally stabilized. Information Builders recommends that you use the ReportCaster Bean API for all scheduling and log functionality.

ReportCaster Servlet API Descriptions

The following table describes the ReportCaster Servlet API.

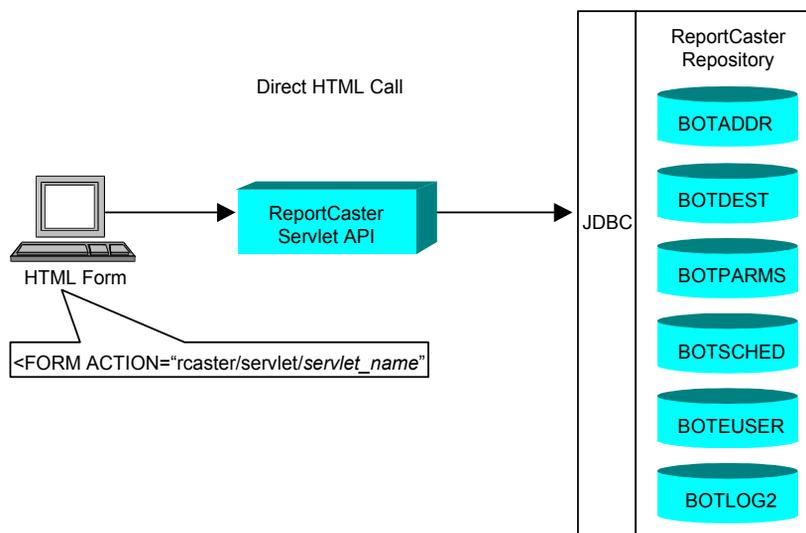
Servlet	Description
DSTDLBULK	Enables the user to create a new Distribution List, add new members to an existing Distribution List, replace the members in a Distribution List, and delete a Distribution List.
DSTDLMEM	Maintains single members in a Distribution List. Enables the user to add a new member or delete an existing member.
DSTDLLIST	Displays a Distribution List in the browser.
DSTSCHED	Schedules the execution and distribution of a report.
DSTACTIVE	Sets the status of a scheduled report, which can be active or inactive. An active report runs and is distributed at the next scheduled interval; an inactive report does not run as scheduled.
DSTRUNNOW	Immediately runs and distributes a report.
DSTLOG	Displays information about the events that occurred during the execution and distribution of a report that was scheduled using the ReportCaster API.

ReportCaster Repository Tables Accessed by the Servlets

The ReportCaster Servlet API writes to, and reads from, a repository that stores distribution and scheduling information. They are either SQL-based (relational) tables, or FOCUS proprietary data sources. The SQL-based version is recommended for applications that have a high volume of scheduled reports.

Note: FOCUS data sources must reside on the same platform as the WebFOCUS Reporting Server.

When the ReportCaster Repository is SQL-based, the ReportCaster Servlet API uses JDBC to write to and read from the repository tables. The following diagram illustrates the communication by the ReportCaster Servlet API to an SQL-based ReportCaster Repository.



When configuring ReportCaster with a FOCUS Repository, the ReportCaster Servlet API connects to the WebFOCUS Reporting Server to communicate to the FOCUS Database Server (FDS) to write to and read from the FOCUS Repository data sources.

Reference ReportCaster Servlet API Repository Tables

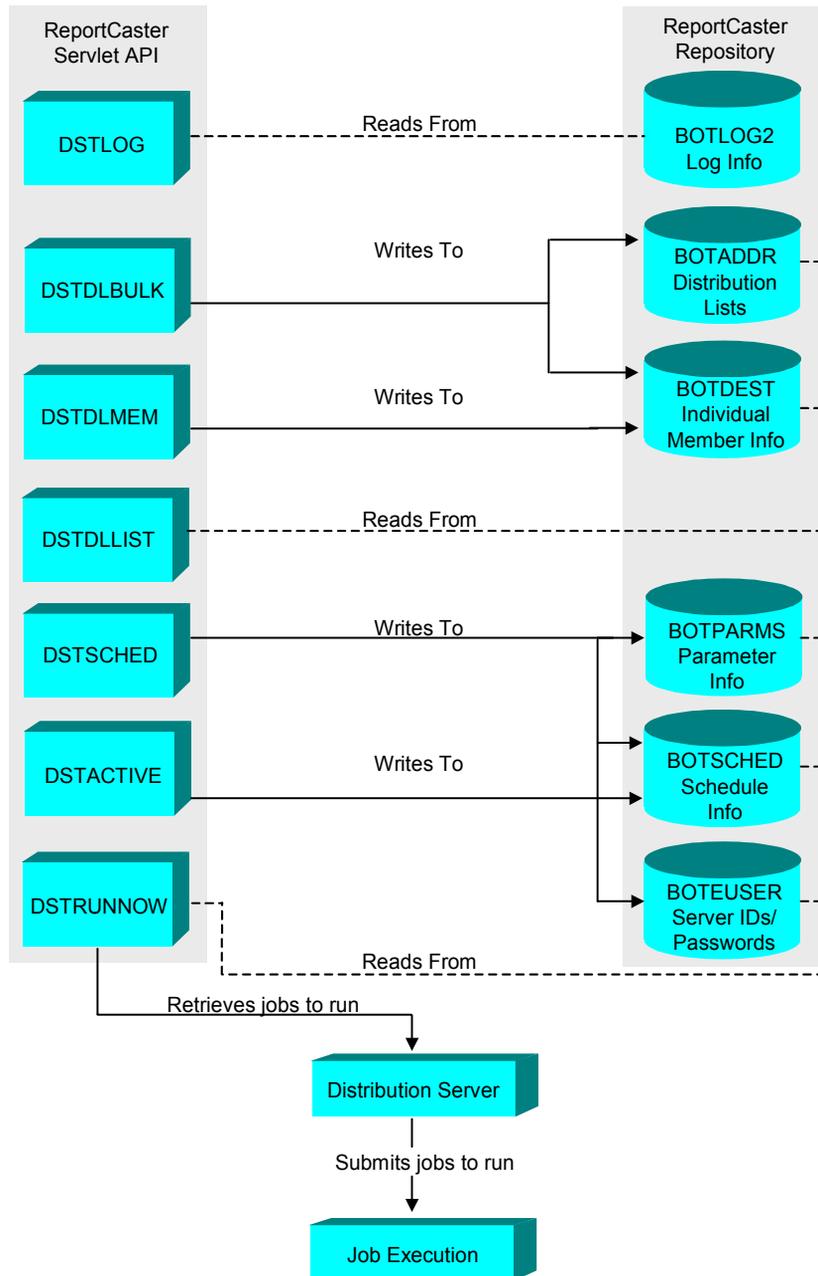
The following ReportCaster Repository tables are accessed by the ReportCaster Servlet API:

Repository Table	Description	Servlet Used
BOTADDR	Contains Distribution Lists, including information such as the distribution method and the owner of the Distribution List.	Maintained by DSTDLBULK. Read by DSTDLLIST.
BOTDEST	Contains information on the individual members of a Distribution List.	Maintained by DSTDLBULK and DSTDLMEM. Read by DSTDLLIST.
BOTPARMS	Contains parameter information. Enables a user to schedule a report to run with certain parameter values.	Maintained by DSTSCHD. Read by DSTRUNNOW.
BOTSCHED	Contains scheduling information, including scheduling interval, procedures that are run before a report, and procedures that are run after a report.	Maintained by DSTSCHD and DSTACTIVE. Read by DSTRUNNOW to immediately run a report.
BOTEUSER	Contains valid WebFOCUS Reporting Server user IDs and passwords.	Maintained by DSTSCHD. Accessed at run time by DSTRUNNOW to obtain the required authorization information.
BOTLOG	Contains log property information (job description, schedule ID, user IDs, and start and end time of the job).	Read by DSTLOG.
BOTLOG2	Contains information on the events related to the execution and distribution of a report. Note: The DSTLOG servlet has the ability to delete records in BOTLOG, which then would delete corresponding records in BOTLOG2.	Read by DSTLOG.

Reference

ReportCaster Servlet API and Repository Tables

The following diagram illustrates the ReportCaster Servlet API and the tables they access in the ReportCaster Repository.



Servlet Security

Before a user can access a ReportCaster Servlet API application, a valid WebFOCUS logon must take place. The ReportCaster Servlet API requires user authentication and configuration information to be established and validated. A WebFOCUS cookie is established when a valid WebFOCUS logon occurs using the WF_SIGNON action. On Windows NT, a sample logon form, rbalogon.htm, is distributed with ReportCaster in `drive:\ibi\WebFOCUSre\ibi_html\broker`

where:

drive

Is the letter of the drive on which WebFOCUS is installed.

rel

Is the WebFOCUS release.

This form navigates to an rbalogon.htm form, which has links to all other sample forms. For more information about sample HTML forms that call ReportCaster Servlet API, see Chapter 6, *ReportCaster API Servlet Samples*.

You can use the supplied logon form, or create your own custom form that assigns the value WF_SIGNON to the variable IBIWF_action. Logging on to the WebFOCUS Reporting Server creates the WebFOCUS cookie (WF_COOKIE) containing the validated user ID of the current user. When the DSTSCHED servlet creates a new schedule, the user ID in the cookie is stored as the owner of the schedule.

The DSTDLBULK and DSTDLMEM servlets, which maintain Distribution Lists, provide a WebFOCUS Reporting Server user ID as a parameter (IBIB_userid) on a transaction. This user ID identifies the owner of the Distribution List and allows users to add records to, and delete records from, Distribution Lists owned by other users (provided that they know the owner's user ID).

In this way, subscription reporting is enabled. An administrator can develop an HTML form that supplies a hidden variable with the owner's user ID of a public Distribution List (for example, `<INPUT TYPE="HIDDEN" NAME="IBIB_userid" VALUE="ownerid">`). Users can then add their e-mail addresses to subscribe to the reports associated with the Distribution List. Listing the contents of a Distribution List (using the DSTDLLIST servlet) is an open function available to all users without identification requirements.

The DSTRUNNOW servlet, which runs a scheduled job immediately, allows users to run their own job or other users' jobs. If the owner's user ID (IBIB_userid) is not supplied, the servlet uses the user ID from the WebFOCUS cookie. If the owner's user ID is supplied, a job associated with that user ID can be run. This feature ensures that the user has knowledge of the owner's user ID before executing their job.

Setting the status of jobs (using the DSTACTIVE servlet) is an open function. The user ID from the WebFOCUS cookie is validated against the user IDs in the ReportCaster Repository. However, only a ReportCaster Administrator user ID can set the status of any job or all jobs in the ReportCaster Repository to active or inactive.

IBIB_userid Parameter

The following table describes the use of the IBIB_userid parameter by each of the servlets:

Servlet	IBIB_userid Description	Security Rule
DSTDLBULK	Required if the userid value in the WeFOCUS cookie is not the owner of the Distribution List, and the user is adding members to an existing Distribution List, replacing members in an existing Distribution List, or deleting an existing Distribution List.	The owner of the scheduled job is the user ID from WF_COOKIE.
	Not used if creating a new Distribution List.	The user ID should not be specified and will be ignored if set.
DSTDLMEM	Optional.	Must supply a value only if the user is not the owner of the Distribution List.
DSTDLLIST	Not applicable.	No security checking. Any user can view any Distribution List.
DSTSCHED	Not applicable.	The owner of the scheduled job is the user ID from WF_COOKIE.
DSTACTIVE	Required if you are not the owner of the schedule.	The user ID from WF_COOKIE is validated against the owner of the schedule. The ReportCaster Administrator ID can set the status of any particular job or for all jobs.
DSTRUNNOW	Required if you are not the owner of the schedule.	The user ID from WF_COOKIE is used. A supplied value overrides the user ID from WF_COOKIE.
DSTLOG	Not applicable.	Only retrieves log reports for jobs that are scheduled through the ReportCaster API and are owned by the current WebFOCUS Reporting Server user ID.

Calling a Servlet From an HTML Form

Call a servlet by:

- Specifying an absolute address for the Web server.
- Specifying a relative address for the Web server.

For details on HTML forms that call specific servlets, see Chapter 6, *ReportCaster API Servlet Samples*.

Syntax

How to Call a Servlet From an HTML Form

Specifying an Absolute Address:

```
<FORM ACTION="http://hostname/rcaster/servlet/servlet_name"
```

where:

hostname

Is the host name of the Web server.

servlet_name

Is the name of the servlet.

Specifying a Relative Address:

```
<FORM ACTION="/rcaster/servlet/servlet_name"
```

where:

servlet_name

Is the name of the servlet.

Servlet Parameters

The HTML calling forms prompt you for parameter values to generate the query string passed to a servlet.

- If you do not supply a required value but a default exists, the servlet will use the default.
- If you do not supply a required value and no default exists, the servlet will return a message.

Reference

ReportCaster Servlet API Schedule Identification Parameters

The ReportCaster Servlet API provides two properties to identify a schedule:

- **IBIB_jobdesc.** This is a user-supplied description that identifies a report request (job) scheduled for execution and distribution. The value must be unique for a given WebFOCUS Reporting Server user ID. The description can be up to 90 alphanumeric characters with embedded blanks and special characters.
- **IBIB_scheduleid.** This is a unique, API-generated, 12-digit ID that identifies a scheduled job in the ReportCaster Repository. It is automatically created when a user schedules a job.

You can identify a job to the ReportCaster API with either the IBIB_jobdesc or the IBIB_scheduleid parameter.

Sample HTML Forms

Sample HTML forms are distributed with ReportCaster in the `ibi_html/broker` subdirectory under the directory where WebFOCUS is installed (the default is `c:\WebFOCUSrel\ibi\` on Windows NT/2000, where *rel* is the WebFOCUS release). You can use these forms as they are, or you can customize them for your application needs. For more information, see Chapter 6, *ReportCaster API Servlet Samples*.

Maintaining and Displaying a Distribution List With a Servlet

The ReportCaster Servlet API enables you to maintain multiple and single members in a Distribution List using the following servlets in a direct HTML call:

- DSTDLBULK to maintain Distribution Lists.
- DSTDLMEM to maintain single Distribution List members.
- DSTDLLIST to display a Distribution List.

Maintaining Distribution Lists Using the DSTDLBULK Servlet

The DSTDLBULK servlet maintains Distribution Lists stored in BOTADDR and BOTDEST. With this servlet you can perform the following functions, using an external Distribution List residing on EDAPATH, or an HTML input form:

- Create a new Distribution List.
- Copy one Distribution List to another.
- Add new members to an existing Distribution List.
- Replace one or more members in an existing Distribution List.
- Delete a Distribution List.

For more information about the rbaulk.htm, rbaulkm.htm, and rbacopy.htm forms, which run the DSTDLBULK servlet, see Chapter 6, *ReportCaster API Servlet Samples*.

Example

Specifying an Absolute Address Using DSTDLBULK

```
<FORM ACTION="http://hostname/rcaster/servlet/DSTDLBULK"
```

where:

hostname

Is the host name of the Web server.

DSTDLBULK Parameters

The following parameters may be used in an HTML calling form to generate the query string passed to the DSTDLBULK servlet.

IBIB_access Parameter for DSTDLBULK

Description	Access allowed on a new Distribution List.
Required?	Yes, if IBIB_function is C, which copies an existing Distribution List to a new Distribution List. Ignored if IBIB_function is N, R, or D.
Size in bytes	2
Valid Values	PR = Private Distribution List (other users cannot access the list). PU = Public Distribution List (anyone can access the list).
Default	PU

IBIB_copy Parameter for DSTDLBULK

Description	Name of source Distribution List. Applies when IBIB_function is one of the following: <ul style="list-style-type: none"> • C, which copies an existing Distribution List to a new Distribution List. • N, which appends an existing Distribution List to another one. • R, which replaces an existing Distribution List with another one.
Required?	Yes, if supplying members from a Distribution List.
Size in bytes	50
Valid Values	An existing Distribution List created using the ReportCaster API.
Default	None

IBIB_filename Parameter for DSTDLBULK

Description	Name of external source file used to supply member names. The file must: <ul style="list-style-type: none">• Reside on EDAPATH (Windows NT and UNIX), or in a partitioned data set (PDS) allocated to ddname EDARPC (OS/390).• Have the extension .fex on Windows NT and UNIX, although you do not specify it. For more information about IBIB_filename, see <i>Creating an External File</i> on page 3-15.
Required?	Yes, if supplying members from an external file when IBIB_function is C, N, or R.
Size in bytes	8
Valid Values	First character must be alphabetic. Single quotation marks and ampersands are not allowed. Blank lines are not allowed in an external distribution file.
Default	None

IBIB_function Parameter for DSTDLBULK

Description	Type of maintenance action that will be performed.
Required?	Yes
Size in bytes	1
Valid Values	C = Create new Distribution List. N = Add new members to existing Distribution List. R = Replace all members in existing Distribution List. D = Delete existing Distribution List.
Default	C

IBIB_method Parameter for DSTDLBULK

Description	Distribution method for a new Distribution List.
Required?	Yes, if IBIB_function is C. Ignored if IBIB_function is N, R, or D.
Size in bytes	5
Valid Values	FTP or F = FTP distribution. MAIL or M = Mail distribution. PRINT or P = Printer distribution.
Default	MAIL

IBIB_name Parameter for DSTDLBULK

Description	Name of the Distribution List that will be modified.
Required?	Yes
Size in bytes	50
Valid Values	An existing Distribution List created using the ReportCaster API.
Default	None

IBIB_recipients Parameter for DSTDLBULK

Description	Records typed into a text box on an HTML form.
Required?	Yes, if supplying members from text box input when IBIB_function is C, N, or R.
Size in bytes	75
Valid Values	A file name (FTP distribution), e-mail address, or printer name. For example: File name ENGLAND.HTM,,,\$ E-mail address Alfred_Stevens@abcd.com,,,\$ Printer name \\WERIBI\29B2.PRNTNY.IBI,,,\$
Default	None

IBIB_tcpiplevel Parameter for DSTDLBULK

Description	Method ReportCaster uses for securing sockets from TCP/IP, based on a specific SAS/C library.
Required?	Yes, if the WebFOCUS Reporting Server runs on OS/390.
Size in bytes	1
Valid Values	1 = Resolves special connectivity problems. 0 = Specifies no special connectivity problems.
Default	0

IBIB_userid Parameter for DSTDLBULK

Description	<p>One of the following:</p> <ul style="list-style-type: none"> User ID of the owner of the target Distribution List when IBIB_copy is not present. The target Distribution List is the one being added to, replaced, or deleted. User ID of the owner of the source Distribution List when IBIB_copy is present. The owner of the target Distribution List is the user ID in the WebFOCUS cookie (WF_COOKIE).
Required?	<p>For adding, replacing, or deleting:</p> <p>A value is required if IBIB_function is N, R, or D. If the WebFOCUS cookie user ID is the owner of the Distribution List, you must still supply a value.</p> <p>For creating:</p> <p>A value is not required if IBIB_function is C. If IBIB_function is C, the user ID from the WebFOCUS cookie is used. If you do supply a value, it will override the ID from the WebFOCUS cookie. This feature enables a logon user (whose ID is stored in the cookie) to create a Distribution List for another user, whose ID is supplied on IBIB_userid.</p>
Size in bytes	48 (Windows NT/2000 and UNIX) 8 (OS/390)
Valid Values	Single quotation marks, ampersands, and spaces are not allowed. First character must be alphabetic (OS/390).
Default	None

Creating an External File

The parameter `IBIB_filename` refers to an external file used to supply member names to a Distribution List. You must provide a value for this parameter when both of the following conditions are true:

- `IBIB_function` is C, N, or R.
- Member names for a Distribution List are supplied from an external file, rather than from a ReportCaster Distribution List or text box input on an HTML form.

When you create an external file, it must contain a maximum of 75-bytes, in comma-delimited format. A record must contain a single destination field for non-bursted report distribution, and an additional burst value field for bursted report distribution.

When creating an external file:

- Separate the fields in each record with a comma.
- Do not exceed the maximum record length of 75 characters.
- Begin each record with the required destination field. Records may or may not contain the optional burst value field.
- Terminate each record with a comma followed by a dollar sign (,\$).
- Do not include blank lines.

Example

Creating an External File for a Bursted Report

For a bursted report, each record in the external file consists of the destination field, followed by the burst value field, and then a comma and a dollar sign to terminate the record. The following are sample records from external files that supply member names to a Distribution List for a bursted report.

The first record illustrates a file name for FTP distribution (`ENGLAND.HTM`). It is followed by the field value on which the report is burst (`ENGLAND`). A comma and a dollar sign terminate the record:

```
ENGLAND.HTM, ENGLAND, $
```

The next record illustrates an e-mail address (`Alfred_Stevens@abcd.com`), followed by the field on which the report is burst (`ENGLAND`). A comma and a dollar sign terminate the record:

```
Alfred_Stevens@abcd.com, ENGLAND, $
```

The last record illustrates a printer name (`\\WERIBI\29B2.PRNTNY.IBI`), followed by the field on which the report is burst (`ENGLAND`). A comma and a dollar sign terminate the record:

```
\\WERIBI\29B2.PRNTNY.IBI, ENGLAND, $
```

Example **Creating an External File for a Non-Bursted Report**

For a non-bursted report, each record in the external file consists of the destination field, followed by a comma to represent the missing burst value field, and then a comma and a dollar sign to terminate the record.

The first sample record illustrates a file name for FTP distribution (ENGLAND.HTM). It is followed by a comma to represent the missing burst value field. The record terminates with a comma and a dollar sign:

```
ENGLAND.HTM, , $
```

The next record illustrates an e-mail address for e-mail distribution. The rest of the record is the same as the preceding record:

```
Alfred_Stevens@abcd.com, , $
```

The last record illustrates a printer name for printer distribution. The rest of the record is the same as the preceding two records:

```
\\WERIBI\29B2.PRNTNY.IBI, , $
```

Generating Text Box Input

The value of the parameter IBIB_recipients results from values typed into a text box on an HTML form. IBIB_recipients is used to supply member names to a Distribution List. You must supply a value for IBIB_recipients when IBIB_function is C, N, or R, and member names for a Distribution List come from an HTML text box rather than from an existing ReportCaster Distribution List or an external file.

IBIB_recipients is composed of instances of the destination field and the burst value field in the same way that IBIB_filename is composed for an external file. For a bursted report, each record consists of the destination field, followed by the burst value field, and then a comma and a dollar sign to terminate the record. For a non-bursted report, each record consists of the destination field, followed by a comma to represent the missing burst value field, and then a comma and a dollar sign to terminate the record.

Creating and Populating a Distribution List Using Text Box Input

You can use the DSTDLBULK servlet to create a new Distribution List and populate it with members from text box input on an HTML form. For more information about the rbaulkm.htm form, which runs the DSTDLBULK servlet, see Chapter 6, *ReportCaster API Servlet Samples*.

This form also enables you to append new members to an existing Distribution List or replace the members in an existing Distribution List.

Maintaining Single Distribution List Members Using the DSTDLMEM Servlet

The DSTDLMEM servlet maintains single members in a Distribution List stored in BOTDEST. With this servlet you can:

- Add a single member to an existing Distribution List.
- Delete a single member from an existing Distribution List.

For more information about the rbadlmem.htm form, which runs the DSTDLMEM servlet, see Chapter 6, *ReportCaster API Servlet Samples*.

Example

Specifying an Absolute Address Using DSTDLMEM

```
<FORM ACTION="http://hostname/rcaster/servlet/DSTDLMEM"
```

where:

hostname

Is the host name of the Web server.

DSTDLMEM Parameters

The following parameters may be used in an HTML calling form to generate the query string passed to the DSTDLMEM servlet.

IBIB_function Parameter for DSTDLMEM

Description	Type of maintenance action that will be performed.
Required?	Yes
Size in bytes	1
Valid Values	A = Add a new member. D = Delete an existing member.
Default	A

IBIB_location Parameter for DSTDLMEM

Description	Destination of the report or report section.
Required?	Yes
Size in bytes	75
Valid Values	File name if IBIB_method is FTP. E-mail address if IBIB_method is MAIL. Printer name if IBIB_method is PRINT
Default	None

IBIB_name Parameter for DSTDLMEM

Description	Name of the Distribution List that will be displayed.
Required?	Yes
Size in bytes	50
Valid Values	An existing Distribution List created using the ReportCaster API.
Default	None

IBIB_tcpiplevel Parameter for DSTDLMEM

Description	Method ReportCaster uses for securing sockets from TCP/IP, based on a specific SAS/C library.
Required?	Yes, if the WebFOCUS Reporting Server runs on OS/390.
Size in bytes	1
Valid Values	1 = Resolves special connectivity problems. 0 = Specifies no special connectivity problems.
Default	0

IBIB_userid Parameter for DSTDLMEM

Description	User ID of the owner of the Distribution List being added to or deleted from.
Required?	Yes
Size in bytes	48 (Windows NT/2000 and UNIX) 8 (OS/390)
Valid Values	Single quotation marks, ampersands, and spaces are not allowed. First character must be alphabetic (OS/390).
Default	None

IBIB_value Parameter for DSTDLMEM

Description	Field value on which the report is burst.
Required?	Yes, if the report is burst.
Size in bytes	75
Valid Values	Burst value
Default	None

Displaying a Distribution List Using the DSTDLLIST Servlet

The DSTDLLIST servlet displays a Distribution List in the browser in HTML format. Any user can view any Distribution List. This information is retrieved from BOTADDR and BOTDEST.

For more information about the rbadlist.htm form, which runs the DSTDLLIST servlet, see Chapter 6, *ReportCaster API Servlet Samples*.

Example

Specifying a Relative Address Using DSTDLLIST

```
<FORM ACTION="/rcaster/servlet/DSTLIST"
```

DSTDLLIST Parameters

The following parameters may be used in an HTML calling form to generate the query string passed to the DSTDLLIST servlet.

IBIB_name Parameter for DSTDLLIST

Description	Name of the Distribution List that will be modified.
Required?	Yes
Size in bytes	50
Valid Values	An existing Distribution List created using the ReportCaster API.
Default	None

IBIB_tcpiplevel Parameter for DSTDLLIST

Description	Method ReportCaster uses for securing sockets from TCP/IP, based on a specific SAS/C library.
Required?	Yes, if the WebFOCUS Reporting Server runs on OS/390.
Size in bytes	1
Valid Values	1 = Resolves special connectivity problems. 0 = Specifies no special connectivity problems.
Default	0

Scheduling a Job Using the DSTSCHED Servlet

The DSTSCHED servlet schedules a new job. It generates a unique, 12-digit schedule ID, which is the key that identifies the new job in BOTSCHED. For more information about the rbasched.htm form, which runs the DSTSCHED servlet, see Chapter 6, *ReportCaster API Servlet Samples*.

By default, the owner of the scheduled job is the user ID from the WebFOCUS cookie.

DSTSCHED Requirements

DSTSCHED is a direct HTML call with following requirements:

- The Distribution List used for the job must exist at the time of scheduling.
- You cannot schedule a report request that resides in a Managed Reporting Repository.
- You cannot direct the output of a scheduled report request to a Prepared Reports folder in a Managed Reporting Repository.

Windows NT/2000 and UNIX only:

- The report request scheduled for execution and distribution must reside in a directory defined on the WebFOCUS Reporting Server, using the environment variable EDAPATH.
- The report request must have the extension .fex.
- A procedure run prior to or after the report request must reside in a directory specified on EDAPATH.

OS/390 only:

The report request scheduled for execution and distribution must reside in a partitioned data set (PDS) allocated to ddname EDARPC.

Example**Specifying a Relative Address Using DSTSCHED**

```
<FORM ACTION="/rcaster/servlet/DSTSCHED">
```

DSTSCHED Parameters

The following parameters may be used in an HTML calling form to generate the query string passed to the DSTSCHED servlet.

IBIB_active Parameter for DSTSCHED

Description	Status of job.
Required?	No
Size in bytes	1
Valid Values	Y = Active. Job will run at next scheduled interval. N = Inactive. Job will not run as scheduled.
Default	Y

IBIB_byfield Parameter for DSTSCHED

Description	Type of report.
Required?	Yes
Size in bytes	1
Valid Values	Y = Report is burst. N = Report is not burst.
Default	N

IBIB_dates Parameter for DSTSCHED

Description	Day of the month the report request will run. This parameter can occur multiple times if IBIB_interval is M (monthly) and the report runs twice or more during the month. Values are required for the specific days in the month, such as the 1st and the 15th. This results in multiple instances of IBIB_dates in the parameter string (for example, IBIB_dates=1, IBIB_dates=15).
Required?	Yes, if IBIB_interval is MONTH (or M). Otherwise, it is ignored.
Size in bytes	2
Valid Values	A number between 1 and 31, or the value Last Day, which means the last day of the month. Internally, Last Day is assigned the value 32 by the servlet.
Default	None

IBIB_distlist Parameter for DSTSCHED

Description	Name of the existing Distribution List containing the recipients of the report. The list must reside in the ReportCaster Repository.
Required?	Yes
Size in bytes	50
Valid Values	An existing Distribution List created using the ReportCaster API.
Default	None

IBIB_enddate Parameter for DSTSCHED

Description	Date of last report execution and distribution, in the format <i>YYYYMMDD</i> , where <i>YYYY</i> is the 4-digit year, <i>MM</i> is the month, and <i>DD</i> is the day of the month.
Required?	No
Size in bytes	8
Valid Values	A date specified as <i>YYYYMMDD</i> .
Default	20990101

IBIB_endtime Parameter for DSTSCHED

Description	Time of last report execution and distribution, in the format <i>HHMM</i> , where <i>HH</i> is the hour and <i>MM</i> is the minute.
Required?	No
Size in bytes	4
Valid Values	A time specified as <i>HHMM</i> .
Default	0000 (midnight)

IBIB_frequency Parameter for DSTSCHED

Description	Number of times the report request is executed and distributed within the specified interval. For example, if you specify a monthly interval (IBIB_interval is M), and you also specify 3 for the number of times (IBIB_frequency is 3), then the report runs every 3 months.
Required?	No
Size in bytes	2
Valid Values	Must be numeric. No special characters or decimals are allowed.
Default	1

IBIB_interval Parameter for DSTSCHED

Description	Period of time (the interval) on which report execution and distribution are based.
Required?	Yes
Size in bytes	5
Valid Values	ONCE or O = One time. HOUR or H = Hourly. DAY or D = Daily. WEEK or W = Weekly. MONTH or M = Monthly. YEAR or Y = Yearly.
Default	ONCE

IBIB_jobdesc Parameter for DSTSCHED

Description	Unique, user-supplied description for the report request (job) being scheduled.
Required?	No
Size in bytes	90
Valid Values	Alphanumeric characters with embedded blanks and special characters are allowed.
Default	Value of IBIB_jobname.

IBIB_jobname Parameter for DSTSCHED

Description	Name of the report request to be scheduled for execution and distribution. It must reside in a path defined to the WebFOCUS Reporting Server. On Windows NT/2000 and UNIX, do not include the extension .fex.
Required?	Yes
Size in bytes	8
Valid Values	Single quotation marks and ampersands are not allowed.
Default	None

IBIB_method Parameter for DSTSCHED

Description	Distribution method.
Required?	Yes
Size in bytes	5
Valid Values	FTP or F = FTP distribution. MAIL or M = Mail distribution. PRINT or P = Printer distribution.
Default	MAIL

IBIB_notifyflag Parameter for DSTSCHED

Description	Flag that controls notification of job status.
Required?	Yes
Size in bytes	1
Valid Values	N = Inactive. No notification is sent. E = Error. Content of the log is sent on an error condition. A = Always. Notification is always sent.
Default	N

IBIB_parm Parameter for DSTSCHED

Description	Values passed to the scheduled job. The values are assigned to variables in the scheduled job and used at run time. For more information on specifying a parameter string, see <i>Passing Values to a Scheduled Job Using DSTSCHED</i> on page 3-33.
Required?	No
Size in bytes	75
Valid Values	Must fit on an 80-byte line, which includes all punctuation.
Default	None

IBIB_postrpc1 and IBIB_postrpc2 Parameters for DSTSCHED

Description	Name of a procedure and optionally its parameter string that will run after the scheduled report request. This procedure typically performs cleanup and related tasks. For example, a value for IBIB_postrpc2 might be: cleanup discard = yes. You can specify up to two procedures.
Required?	No
Size in bytes	65
Valid Values	An existing server procedure residing on EDAPATH.
Default	None

IBIB_prerpc1 and IBIB_prerpc2 Parameters for DSTSCHED

Description	Name of a procedure and optionally its parameter string that will run prior to the scheduled report request. This procedure typically performs setup tasks. For example, a value for IBIB_prerpc1 might be: setup region = west. You can specify up to two procedures.
Required?	No
Size in bytes	65
Valid Values	An existing server procedure residing on EDAPATH.
Default	None

IBIB_sendformat Parameter for DSTSCHED

Description	Format of the distributed report output.
Required?	Yes
Size in bytes	8
Valid Values	HTML = Hypertext Markup Language. PDF = Adobe Acrobat Portable Document Format. PS = PostScript. DOC = Text with ASCII form feeds. WP = Text (non-formatted for multiple word processing applications). EXCEL = Excel spreadsheet. EXL2K = Excel 2000 spreadsheet. WK1 = Lotus 1-2-3 spreadsheet.
Default	HTML

IBIB_startdate Parameter for DSTSCHED

Description	Date of first report execution and distribution, in the format <i>YYYYMMDD</i> , where <i>YYYY</i> is the 4-digit year, <i>MM</i> is the month, and <i>DD</i> is the day of the month.
Required?	No
Size in bytes	8
Valid Values	A date specified as <i>YYYYMMDD</i> .
Default	Current system date.

IBIB_starttime Parameter for DSTSCHED

Description	Time of first report execution and distribution, in the format <i>HHMM</i> , where <i>HH</i> is the hour and <i>MM</i> is the minute.
Required?	No
Size in bytes	4
Valid Values	A time specified as <i>HHMM</i> .
Default	Current system time.

IBIB_tcpiplevel Parameter for DSTSCHED

Description	Method ReportCaster uses for securing sockets from TCP/IP, based on a specific SAS/C library.
Required?	Yes, if the WebFOCUS Reporting Server runs on OS/390.
Size in bytes	1
Valid Values	1 = Resolves special connectivity problems. 0 = Specifies no special connectivity problems.
Default	0

IBIB_weekdays Parameter for DSTSCHED

Description	Day of the week the report request will run. This parameter can occur multiple times. For example, if you specify a weekly interval (IBIB_interval is W), and you also specify 3 for the number of times (IBIB_frequency is 3), then the report runs every 3 weeks. Values are required for the specific days of the week (such as Monday, Wednesday, and Friday), resulting in multiple parameters in the query string: weekdays=MON, weekdays=WED, weekdays=FRI.
Required?	Yes, if IBIB_interval is WEEK (or W). If IBIB_interval is DAY (or D), and a value is not supplied, the report runs every day of the week. In all other cases, this parameter is ignored.
Size in bytes	5
Valid Values	MON = Monday. TUES = Tuesday. WED = Wednesday. THURS = Thursday. FRI = Friday. SAT= Saturday. SUN = Sunday.
Default	None

Mail Parameters for DSTSCHED

The following parameters apply only if IBIB_method is MAIL (or M).

IBIB_mailcompany Parameter for DSTSCHED

Description	The company to which the sender is associated.
Required?	No
Size in bytes	64
Valid Values	Single quotation marks and ampersands are not allowed.
Default	None

IBIB_mailfrom Parameter for DSTSCHED

Description	E-mail address for return mail (the reply address). DSTSCHED checks the value for the presence of the character @ and returns a message if it is not there.
Required?	No
Size in bytes	64
Valid Values	Single quotation marks and ampersands are not allowed. An @ symbol must be included.
Default	None

IBIB_mailhost Parameter for DSTSCHED

Description	Name of the mail server that distributes the report.
Required?	Yes, if IBIB_method is MAIL (or M).
Size in bytes	64
Valid Values	Single quotation marks and ampersands are not allowed.
Default	Value specified in the ReportCaster configuration file.

IBIB_mailsubject Parameter for DSTSCHED

Description	Text that describes the content or purpose of the e-mail.
Required?	No
Size in bytes	64
Valid Values	Single quotation marks and ampersands are not allowed.
Default	None

FTP Parameters for DSTSCHED

The following parameters apply only if IBIB_method is FTP (or F).

IBIB_asvalue Parameter for DSTSCHED

Description	Name of the FTP index file for a bursted report.
Required?	No
Size in bytes	8
Valid Values	First character must be alphabetic.
Default	HOLD

IBIB_ftphost Parameter for DSTSCHED

Description	Name of FTP server that distributes the report.
Required?	Yes, if IBIB_method is FTP (or F).
Size in bytes	64
Valid Values	Single quotation marks and ampersands are not allowed.
Default	Value specified in the ReportCaster configuration file.

IBIB_ftplocation Parameter for DSTSCHED

Description	Destination of the FTP-distributed report. It must be a subdirectory of the FTP server root directory, or an alias defined to the FTP server.
Required?	Yes, if IBIB_method is FTP (or F).
Size in bytes	64
Valid Values	Single quotation marks and ampersands are not allowed.
Default	Value specified in the ReportCaster configuration file.

IBIB_ftppass Parameter for DSTSCHED

Description	Password associated with the FTP user ID.
Required?	Yes, if IBIB_method is FTP (or F).
Size in bytes	64
Valid Values	Spaces are not allowed.
Default	Value specified in the ReportCaster configuration file.

IBIB_ftpuser Parameter for DSTSCHED

Description	User ID authorized for FTP transfer.
Required?	Yes, if IBIB_method is FTP (or F).
Size in bytes	64
Valid Values	Single quotation marks and ampersands are not allowed.
Default	Value specified in the ReportCaster configuration file.

Notification Parameters for DSTSCHED

The following parameters apply only if IBIB_notifyflag is E or A.

IBIB_notifyaddress Parameter for DSTSCHED

Description	E-mail address of the person who will receive full notification (complete log report). DSTSCHED checks the value for the presence of the character @ and returns a message if it is not there.
Required?	Yes, if IBIB_notifyflag is E or A and IBIB_notifybrief is not supplied. May also be supplied if IBIB_notifybrief is present.
Size in bytes	64
Valid Values	Single quotation marks and ampersands are not allowed. An @ symbol must be included.
Default	None

IBIB_notifybrief Parameter for DSTSCHED

Description	E-mail address of cell phone, pager, or other hand-held device that will receive the abbreviated notification (job schedule ID, job description, and brief status messages).
Required?	Yes, if IBIB_notifyflag is E or A and IBIB_notifyaddress is not supplied. May also be supplied if IBIB_notifyaddress is present.
Size in bytes	64
Valid Values	No restrictions.
Default	None

IBIB_notifyreply Parameter for DSTSCHED

Description	E-mail address for return mail (the response to the notification message). DSTSCHED checks the value for the presence of the character @ and returns an error message if it is not there.
Required?	Yes, if IBIB_notifyflag is E or A.
Size in bytes	64
Valid Values	Single quotation marks and ampersands are not allowed. An @ symbol must be included.
Default	None

IBIB_notifysubject Parameter for DSTSCHED

Description	Text that describes the content or purpose of the notification message.
Required?	No
Size in bytes	90
Valid Values	Single quotation marks and ampersands are not allowed.
Default	Value of IBIB_jobdesc.

Passing Values to a Scheduled Job Using DSTSCHED

With DSTSCHED, you can pass values to the job that you are scheduling. The values are assigned to the appropriate variables in the scheduled job and used at run time.

You can pass an unlimited number of values to a scheduled job. The values are stored in the BOTPARMS table in the ReportCaster Repository.

Use one of the following formats to pass values to a scheduled job:

- Append the variable name used in the scheduled job to IBIB_parm, using an underscore as the delimiter.
- Set IBIB_parm to the full assignment statement. This format uses an equal sign.

Syntax

How to Pass Values to a Scheduled Job Using a Variable Within the Parameter

```
IBIB_parm_variable_name='variable_value'
```

where:

```
IBIB_parm_variable_name
```

Is the composite name of the parameter (IBIB_parm) and the Dialogue Manager variable in the scheduled job (*variable_name*).

```
variable_value
```

Is the value for the Dialogue Manager variable. Embedded blanks must be enclosed in single quotation marks.

Example

Passing a Value to a Report Request Using a Variable Within the Parameter

Consider the following report request, which contains the Dialogue Manager amper variable &TITLE:

```
TABLE FILE MOVIES
PRINT DIRECTOR CATEGORY
BY TITLE
WHERE TITLE IS '&TITLE'
END
```

The HTML calling form prompts the user for a value for IBIB_parm_TITLE. Assume that the user enters 'DOG DAY AFTERNOON'. The report request, when run, selects only the record for that movie:

<u>TITLE</u>	<u>DIRECTOR</u>	<u>CATEGORY</u>
DOG DAY AFTERNOON	LUMET S.	DRAMA

The following example illustrates the format used on an HTML form to pass the value to the request:

```
<input type="text" name="IBIB_parm_TITLE" value="DOG DAY AFTERNOON"
```

The query string assignment that results is:

```
IBIB_parm_TITLE='DOG DAY AFTERNOON'
```

Syntax

How to Pass Values to a Scheduled Job Using a Full Assignment Statement

```
IBIB_parm='variable_name=variable_value'
```

where:

variable_name

Is the name of the Dialogue Manager variable in the scheduled job.

variable_value

Is the value for the Dialogue Manager variable. Embedded blanks must be enclosed in single quotation marks.

Example

Passing a Value to a Report Request Using a Full Assignment Statement

Consider the following report request, which contains the Dialogue Manager amper variable &TITLE:

```
TABLE FILE MOVIES
PRINT DIRECTOR CATEGORY
BY TITLE
WHERE TITLE IS '&TITLE'
END
```

The HTML calling form prompts the user for a value for IBIB_parm_TITLE. Assume that the user enters 'DOG DAY AFTERNOON'. The report request, when run, selects only the record for that movie:

```
TITLE                  DIRECTOR  CATEGORY
DOG DAY AFTERNOON  LUMET S.   DRAMA
```

The following example illustrates the format used on an HTML form to pass the value to the request:

```
<input type="text" name="IBIB_parm" value="TITLE=DOG DAY AFTERNOON"
```

The query string assignment that results is:

```
IBIB_parm='TITLE=DOG DAY AFTERNOON'
```

Setting the Status of a Job Using the DSTACTIVE Servlet

The DSTACTIVE servlet is a direct HTML call that sets the status of a job. With this servlet, you can set the status of:

- A particular job to active or inactive.
- Any job or all jobs in the BOTSCHED table to active or inactive if you are logged on to the application with a valid ReportCaster Administrator user ID.

When you set a status, the IBIB_active flag in the BOTSCHED table changes to reflect that status. For more information about the rbastats.htm form, which runs the DSTACTIVE servlet, see Chapter 6, *ReportCaster API Servlet Samples*.

Reference

DSTACTIVE Requirements

- If there are multiple schedules for a report request (job), you must supply the parameter IBIB_scheduleid. A ReportCaster Administrator can determine a schedule ID for a job by running the report from the ReportCaster Console. Both administrators and users can determine a schedule ID from the Log File icon on the ReportCaster toolbar.
- If there is one schedule for a job, supply either IBIB_scheduleid or IBIB_jobdesc.

Example

Specifying a Relative Address Using DSTACTIVE

```
<FORM ACTION="/rcaster/servlet/DSTACTIVE"
```

DSTACTIVE Parameters

The following parameters may be used in an HTML calling form to generate the query string passed to the DSTACTIVE servlet.

IBIB_active Parameter for DSTACTIVE

Description	Flag that reflects the status of a job.
Required?	Yes
Size in bytes	1
Valid Values	Y = Job is active and will run at next scheduled interval. N = Job is inactive and will not run at next scheduled interval.
Default	Y

IBIB_jobdesc Parameter for DSTACTIVE

Description	Unique, user-supplied description that identifies a job that was scheduled for execution and distribution.
Required?	Yes, if you scheduled a job to run once and IBIB_scheduleid is not supplied.
Size in bytes	90
Valid Values	<ul style="list-style-type: none"> Entering ALL updates the status of all jobs in BOTSCHEID (requires ReportCaster Administrator user ID). <p>or</p> <ul style="list-style-type: none"> An existing description used for schedule created by the ReportCaster API.
Default	None

IBIB_scheduleid Parameter for DSTACTIVE

Description	Unique, API-generated key that identifies a scheduled job.
Required?	Yes, if there are multiple schedules for a job. If you scheduled a job to run once, either this parameter or IBIB_jobdesc is required.
Size in bytes	12
Valid Values	<ul style="list-style-type: none"> Entering ALL updates the status of all jobs in BOTSCHEID (requires ReportCaster Administrator user ID). <p>or</p> <ul style="list-style-type: none"> An existing schedule ID generated by the ReportCaster API.
Default	None

IBIB_tcpiplevel Parameter for DSTACTIVE

Description	Method ReportCaster uses for securing sockets from TCP/IP, based on a specific SAS/C library.
Required?	Yes, if the WebFOCUS Reporting Server runs on OS/390.
Size in bytes	1
Valid Values	1 = Resolves special connectivity problems. 0 = Specifies no special connectivity problems.
Default	0

Immediately Running a Scheduled Job Using the DSTRUNNOW Servlet

The DSTRUNNOW servlet is a direct HTML call that runs a scheduled job immediately, regardless of its status (active or inactive) in the ReportCaster Repository. The job runs as soon as a WebFOCUS Reporting Server agent is available.

With this servlet you can pass values to the scheduled job you are running. The values are assigned to the appropriate variables in the scheduled job and used at run time. For syntax and examples about passing values to a scheduled job, see *Passing Values to a Scheduled Job Using DSTSCHED* on page 3-33, making sure to substitute DSTRUNNOW for every instance of DSTSCHED.

For more information about the rbarunow.htm form, which runs the DSTRUNNOW servlet, see Chapter 6, *ReportCaster API Servlet Samples*.

Reference

DSTRUNNOW Requirements

- The Distribution List used for the job must already exist.
- The schedule information for the job must already exist. A ReportCaster Administrator can determine a schedule ID for a job by running the report from the ReportCaster Console. Both administrators and users can determine a schedule ID from the Log File icon on the ReportCaster toolbar.
- The call must be made from a secure environment. This means that the user must log on to your application with a valid WebFOCUS Reporting Server user ID and password, which is then written to a WebFOCUS cookie and accessed by the API. The user ID for the server in the cookie must match the user ID stored in the ReportCaster Repository for the target job schedule.
- To run another user's job, supply their user ID for the IBIB_userid parameter.
- If there are multiple schedules for a report request (job), you must supply the parameter IBIB_scheduleid.
- If there is one schedule for a job, you may supply either IBIB_scheduleid or IBIB_jobdesc.

Example

Specifying a Relative Address Using DSTRUNNOW

```
<FORM ACTION="/rcaster/servlet/DSTRUNNOW"
```

DSTRUNNOW Parameters

The following parameters may be used in an HTML calling form to generate the query string passed to the DSTRUNNOW servlet.

IBIB_jobdesc Parameter for DSTRUNNOW

Description	Unique, user-supplied description for the scheduled job.
Required?	Yes, if the job was scheduled to run once and IBIB_scheduleid is not supplied.
Size in bytes	90
Valid Values	Existing description used for schedule created by the ReportCaster API.
Default	None

IBIB_parm Parameter for DSTRUNNOW

Description	Values passed to the scheduled job. The values are assigned to variables in the scheduled job and used at run time. For more information on specifying a parameter string, see <i>Passing Values to a Scheduled Job Using DSTSCHED</i> on page 3-33.
Required?	No
Size in bytes	75
Valid Values	Must fit on an 80-byte line, which includes all punctuation.
Default	None

IBIB_priority Parameter for DSTRUNNOW

Description	Priority level for the job scheduled to run.
Required?	No
Size in bytes	1
Valid Values	1 = Highest priority 2 = Class 2 priority 3 = Class 3 priority 4 = Class 4 priority 5 = Lowest priority
Default	3

IBIB_scheduleid Parameter for DSTRUNNOW

Description	Unique, API-generated key that identifies the scheduled job.
Required?	Yes, if there are multiple schedules for a job. If the job was scheduled to run once, either this parameter or IBIB_jobdesc is required.
Size in bytes	12
Valid Values	Existing schedule ID generated by the ReportCaster API.
Default	None

IBIB_tcpiplevel Parameter for DSTRUNNOW

Description	Method ReportCaster uses for securing sockets from TCP/IP, based on a specific SAS/C library.
Required?	Yes, if the WebFOCUS Reporting Server runs on OS/390.
Size in bytes	1
Valid Values	1 = Resolves special connectivity problems. 0 = Specifies no special connectivity problems.
Default	0

IBIB_userid Parameter for DSTRUNNOW

Description	User ID of the owner of the scheduled job.
Required?	No. If you do not supply a value, DSTRUNNOW uses the user ID from the WebFOCUS cookie. If you do supply a value, it will override the user ID from the WebFOCUS cookie. This feature enables a logged on user (whose ID is stored in the cookie) to run a job owned by another user (whose ID is supplied for IBIB_userid).
Size in bytes	48 (Windows NT and UNIX) 8 (OS/390)
Valid Values	Single quotation marks, ampersands, and spaces are not allowed.
Default	User ID from the WebFOCUS cookie.

Displaying Log Information Using the DSTLOG Servlet

The DSTLOG servlet displays information about the events that occurred during the execution and distribution of a report. It enables you to confirm that a report job ran as scheduled and was distributed successfully. If it did not run or was not distributed successfully, the log information states the reason why.

DSTLOG extracts information from the ReportCaster Repository tables and displays it in your browser in HTML format. You can select specific information for display by supplying any combination of job description, schedule ID, last job execution, and date/time.

If you do not supply any selection criteria, all log information is retrieved and displayed.

Note: You can only view log reports for schedules created with the ReportCaster API when using the DSTLOG servlet. Otherwise, the following message displays:

No Log information exists for this request.

For more information about the rbalog.htm form, which runs the DSTLOG servlet, see Chapter 6, *ReportCaster API Servlet Samples*.

Example

Specifying a Relative Address Using DSTLOG

```
<FORM ACTION="/rcaster/servlet/DSTLOG">
```

DSTLOG Parameters

The following parameters may be used in an HTML calling form to generate the query string passed to the DSTLOG servlet.

IBIB_enddate Parameter for DSTLOG

Description	Ending date for collection of log information, in the format <i>YYYYMMDD</i> , where <i>YYYY</i> is the 4-digit year, <i>MM</i> is the month, and <i>DD</i> is the day of the month. If you supply a value, log information up to that date will be displayed.
Required?	No
Size in bytes	8
Valid Values	A date specified as <i>YYYYMMDD</i> .
Default	None

IBIB_endtime Parameter for DSTLOG

Description	Ending time for collection of log information, in the format <i>HHMM</i> , where <i>HH</i> is the hour and <i>MM</i> is the minute. If you supply a value, log information up to that time will be displayed.
Required?	No
Size in bytes	4
Valid Values	A time specified as <i>HHMM</i> .
Default	2300

IBIB_jobdesc Parameter for DSTLOG

Description	Unique, user-supplied description for a scheduled job. If you supply a value, log information for every execution of that job will be displayed.
Required?	No
Size in bytes	90
Valid Values	Existing description used for schedule created by the ReportCaster API.
Default	None

IBIB_lastexec Parameter for DSTLOG

Description	Flag that controls which job executions will display. This parameter applies to jobs that were executed more than once.
Required?	No
Size in bytes	1
Valid Values	Y = Display last execution of specified job. N = Display all instances that meet other specified criteria.
Default	None

IBIB_scheduleid Parameter for DSTLOG

Description	Unique, API-generated key that identifies a scheduled job. If you supply a value, log information for that schedule ID will be displayed.
Required?	No
Size in bytes	12
Valid Values	Existing schedule ID generated by the ReportCaster API.
Default	None

IBIB_startdate Parameter for DSTLOG

Description	Starting date for collection of log information, in the format <i>YYYYMMDD</i> , where <i>YYYY</i> is the 4-digit year, <i>MM</i> is the month, and <i>DD</i> is the day of the month. If you supply a value, log information starting on that date will be displayed.
Required?	No
Size in bytes	8
Valid Values	A date specified as <i>YYYYMMDD</i> .
Default	None

IBIB_starttime Parameter for DSTLOG

Description	Starting time for collection of log information, in the format <i>HHMM</i> , where <i>HH</i> is the hour and <i>MM</i> is the minute. If you supply a value, log information starting at that time will be displayed.
Required?	No
Size in bytes	4
Valid Values	A time specified as <i>HHMM</i> .
Default	0000

IBIB_tcpiplevel Parameter for DSTLOG

Description	Method ReportCaster uses for securing sockets from TCP/IP, based on a specific SAS/C library.
Required?	Yes, if the WebFOCUS Reporting Server runs on OS/390.
Size in bytes	1
Valid Values	1 = Resolves special connectivity problems. 0 = Specifies no special connectivity problems.
Default	0

Log Content

Log information includes the following:

- **Job Description.** The unique description that the user supplied to identify the report request (job) when it was scheduled.
- **Server User.** WebFOCUS Reporting Server user ID, indicating the owner of the job.
- **MRE User.** Managed Reporting user ID, indicating the owner of the job. A message informs you if this field does not apply.
- **Process.** A unique, API-generated key that identifies a specific execution of the scheduled job.
- **Procedure.** The name of the report request scheduled for execution and distribution. This field contains the name of a procedure stored on the WebFOCUS Reporting Server (for example: acctrpt).
- **Schedule ID.** A unique, API-generated, 12-digit key assigned to the job when it was scheduled.
- **Start Time.** The date and time the job started running.
- **End Time.** The date and time the job finished running.
- **Messages.** These consist of:
 - General information, such as the method of distribution for a particular job (for example, mail distribution).
 - Processing information, indicating that the request started, distribution was successful, and the request completed. Processing information also includes reasons why a request failed, such as unavailability of a data source.

The following is a sample ReportCaster log report.

Job Process Log Report	
Job Description: Sales by State	
Server User: radmin	(BTP1010) Starting worker thread
MRE User: admin	(BTP1010) Method: Mail. Host: 127.0.0.1.
Process: P0t74ok9f01	(BTX000) (FOC43011) FILE hold.htm SUCCESSFULLY DISTRIBUTED
Procedure: app/salesbys	(BTP3080) Resolving Broker Server temporary space for BKRLOG service
Schedule ID: St74obptk1	(BTP3081) Temporary directory for BKRLOG process follows on next line
Start Time: 2001-10-26 11:04:17	(BTP3081) C:\ibi\srvt436\wfs\datemp\ts000216\
End Time: 2001-10-26 11:05:58	(BTY1010) Broker Request app/salesbys Complete.
Job Description: East Coast Sales Report	
Server User: radmin	(BTP1010) Starting worker thread
MRE User: admin	(BTP1010) Method: Mail. Host: 127.0.0.1.
Process: P0t74onvpc2	(BTX000) (FOC43011) FILE hold.htm SUCCESSFULLY DISTRIBUTED
Procedure: app/salesbyc	(BTP3080) Resolving Broker Server temporary space for BKRLOG service
Schedule ID: St74of0sm2	(BTP3081) Temporary directory for BKRLOG process follows on next line
Start Time: 2001-10-26 11:06:16	(BTP3081) C:\ibi\srvt436\wfs\datemp\ts000216\
End Time: 2001-10-26 11:08:05	(BTY1010) Broker Request app/salesbyc Complete.

CHAPTER 4

ReportCaster API Subroutines

Topics:

- API Subroutines
- Tables Accessed by the Subroutines
- Subroutine Security
- Calling a Subroutine From a Procedure
- Maintaining Distribution Lists Using the DSTBULK Subroutine
- Maintaining Single Distribution List Members Using the DSTMEM Subroutine
- Running a Scheduled Job Using the DSTRUN Subroutine
- Using Amper Variables Within a Subroutine

ReportCaster API subroutines are a set of C-based API functions that call a servlet. The subroutines are called from a procedure, and they enable a user to:

- Create and maintain Distribution Lists that contain the addresses of recipients of scheduled reports. These lists may be public or private Distribution Lists.
- Immediately run and distribute a report.

API Subroutines

The following table describes the API subroutines, which are automatically installed with the WebFOCUS Reporting Server when ReportCaster is installed.

API Subroutine	Description
DSTBULK	Enables the user to create a new Distribution List, add new members to an existing Distribution List, replace the members in a Distribution List, and delete a Distribution List.
DSTMEM	Maintains single members in a Distribution List. Enables the user to add a new member or delete an existing member.
DSTRUN	Immediately runs and distributes a report.

Tables Accessed by the Subroutines

The ReportCaster API subroutines write to, and read from, a repository that stores distribution and scheduling information. They are either SQL-based (relational) tables, or FOCUS proprietary data sources. The SQL-based version is recommended for applications that have a high volume of scheduled reports.

Note: FOCUS data sources must reside on the same platform as the WebFOCUS Reporting Server.

The following ReportCaster Repository tables are accessed by the API subroutines.

Repository Table	Description	Subroutine Used
BOTADDR	Contains Distribution Lists with descriptive information, such as distribution method and access allowed (public or private).	Maintained by DSTBULK.
BOTDEST	Contains information on the individual members of a Distribution List.	Maintained by DSTBULK and DSTMEM.
BOTPARMS	Contains parameter information. Enables a user to schedule a report to run with certain parameter values.	Read by DSTRUN.
BOTSCHED	Contains scheduling information, including scheduling interval, procedures that are run before a report, and procedures that are run after a report.	Read by DSTRUN to immediately run a report.
BOTEUSER	Contains valid WebFOCUS Reporting Server user IDs and passwords.	Read by DSTRUN at run time to obtain the required authorization information.

Subroutine Security

All ReportCaster API subroutines require a WebFOCUS Reporting Server user ID and password as the first and third arguments. The subroutines use this user ID and password internally to perform a WebFOCUS logon.

ReportCaster uses the WebFOCUS Reporting Server user ID to determine ownership of schedules, Distribution Lists, and log files. Your WebFOCUS Reporting Server user ID is stored in the ReportCaster Repository in the case you specify at logon. You must specify your WebFOCUS Reporting Server user ID and password in the same case each time you access ReportCaster to obtain access to all your ReportCaster schedules, Distribution Lists, and log files.

Other security considerations are:

- The DSTBULK and DSTMEM subroutines require that an owner's user ID for a target Distribution List be passed on a transaction that creates or manipulates the list.
- If the Web server is running with security on, the *httpuser/pswd* and *httpuser/pswd_length* subroutine arguments are required.
- The DSTRUN subroutine, which runs a scheduled job immediately, allows the user to run their own schedule or other users' schedules. When running another user's schedule, the owner's user ID must be supplied. This ensures that the user has knowledge of the owner's user ID before executing his job.

Calling a Subroutine From a Procedure

You may call a ReportCaster API subroutine from:

- A scheduled procedure using a -SET Dialogue Manager command. This can also be called as a pre-processing or post-processing step.
- A procedure called by an HTML form.
- A procedure run in batch processing on the WebFOCUS Reporting Server.

A procedure must reside in a path defined to the WebFOCUS Reporting Server. On Windows NT/2000 and UNIX, it must be located in EDAPATH. On OS/390, it must be in a partitioned data set (PDS) allocated to ddname EDARPC.

Calling an API Subroutine Using -SET

You can call an API subroutine using a -SET Dialogue Manager command. For more information about Dialogue Manager, see your *WebFOCUS Developing Reporting Applications* manual.

Syntax

How to Call an API Subroutine Using -SET

A ReportCaster API subroutine passes a required set of arguments in a specified order and supplies a return code to the calling procedure. You can change the Dialogue Manager variable names (amper variables), or replace them with coded values. The following variable names used are samples.

Note: This is the general syntax for all subroutines. For more information about the syntax for each subroutine, see the specific sections that describe the DSTBULK, DSTMEM, and DSTRUN subroutines.

```
-SET &var_name=API_subrtn_name(srv_userid,srv_userid_length,  
- 'srv_userpass',srv_userpass_length,  
- 'hostname_port',hostname_port_length,  
- 'input7',  
- 'input8',  
. . .  
- 'httpuser/pwd',httpuser/pwd_length,  
[- 'tcpiplevel',]  
- 'I4');
```

where:

&var_name

Is the variable that will contain the return code.

API_subrtn_name

Is the API subroutine, which can be DSTBULK, DSTMEM, or DSTRUN.

srv_userid

Is a valid WebFOCUS Reporting Server user ID. It must match the user ID stored in the ReportCaster Repository with the job information.

srv_userid_length

Is the length of the WebFOCUS Reporting Server user ID.

srv_userpass

Is a valid password for the user ID. It must match the password of the user ID stored in the ReportCaster Repository with the job information

srv_userpass_length

Is the length of the WebFOCUS Reporting Server user password.

hostname_port

Is the host name (or IP address) and port number of the Web server on which the WebFOCUS Client is installed.

If the port number is 80 (the default), you may omit the colon and port number. For example,

'hostname',

If the port number is not 80, use a colon as a delimiter and then specify the port number. For example,

'hostname:81',

hostname_port_length

Is the length of the host name and port number. This will be only the length of the host name if the port number is omitted.

input7, input8, ...

Are the input arguments (the values required by the subroutine). These arguments are different for each subroutine.

httpuser/pswd

Is a valid user ID and password for the Web server, separated by the character /. A user ID and password are required if the Web server is running with security on.

httpuser/pswd_length

Is the length of the Web server user ID and password, including the character /.

tcpiplevel

Required only if the WebFOCUS Reporting Server runs on OS/390. This argument is the method the ReportCaster API uses for securing sockets from TCP/IP, based on a specific SAS/C library. Valid values are:

1
Resolves special connectivity problems.

0
Indicates that there are no special connectivity problems.

'I4'

Is the format of the return code. For more information, see Appendix B, *ReportCaster API Messages and Codes*.

Reference

Requirements for Coding an API Subroutine Using -SET

When you code a ReportCaster API subroutine:

- You must code the arguments passed to the API subroutine in the exact order as shown in *How to Call an API Subroutine Using -SET* on page 4-4. The arguments are positional.

If you do not supply all arguments, the subroutine will terminate, and a TSCOM3 message may be generated.

The only exception is the argument *tcpipllevel*, which is required only if the WebFOCUS Reporting Server runs on OS/390. If your server does not run on OS/390, you must omit this argument.

- If you do not supply a value for an alphanumeric argument that is optional, you must code the following to hold the position for the argument and to indicate a length of zero. There is one space between the single quotation marks:

```
' ',0,
```

- You must code a comma to separate arguments.
- You must code alphanumeric values in single quotation marks. For example, if the name of the WebFOCUS Reporting Server is IBIWNT and the port number is 1234, you must code the value as 'IBIWNT:1234'.
- You must code a continuation line with a hyphen (-) followed by one or more spaces.
- A line in a procedure cannot exceed 80 characters. Since amper variables are expanded when assigned their value, code associated pairs of variables (for example, *srv_userpass*, *srv_userpass_length*) on separate lines from other variables to ensure that you do not exceed the maximum line length.

Note:

- WordPad (at times) places blank spaces at the end of lines that WebFOCUS does not parse properly. This is more likely to happen if you cut and paste from a document.
- Using WordPad to edit a -SET command with arguments for a subroutine can result in the following message:

```
Unbalanced Parenthesis
```

Reference**Identification Arguments for ReportCaster API Subroutines**

The ReportCaster API provides a pair of identification arguments as input to certain subroutines.

- **jobdesc.** This is a user-supplied description that identifies a report request (job) scheduled for execution and distribution. The value must be unique for a given WebFOCUS Reporting Server user ID. The description can be up to 90 alphanumeric characters with embedded blanks and special characters.
- **scheduleid.** This is a unique, API-generated, 12-digit ID that identifies a scheduled job in the ReportCaster Repository. It is automatically created when a user schedules a job.

You can identify a job to the ReportCaster API with either the jobdesc or the scheduleid argument.

Note: The ReportCaster Administrator can determine a schedule ID for a job by running the report from the ReportCaster Console. Both administrators and users can determine a schedule ID by using the Log File icon on gray toolbar of the ReportCaster Interface.

Syntax**How to Code an HTML Form That Calls a Procedure**

You can call a procedure from an HTML form. The following sample HTML code calls a procedure, which in turn calls a ReportCaster API subroutine. For more details on coding syntax, see your *WebFOCUS Developing Reporting Applications* manual, and the HTML files residing on `/ibi_html/broker/`.

```
<FORM ACTION="/cgi-bin/ibi_cgi/ibiweb.exe" METHOD="get">
<INPUT NAME="IBIF_ex" VALUE="procedure_name" TYPE="hidden">
```

where:

procedure_name

Is the name of the procedure to run. It contains the call to the ReportCaster API subroutine. A procedure must reside in a path defined to the WebFOCUS Reporting Server. On Windows NT/2000 and UNIX, it must be located in EDAPATH. On OS/390, it must be in a partitioned data set (PDS) allocated to ddname EDARPC. Do not specify the file name extension in the VALUE= field.

Maintaining Distribution Lists Using the DSTBULK Subroutine

The DSTBULK subroutine maintains Distribution Lists stored in BOTADDR and BOTDEST. It is a procedure-based call. With this subroutine, you can perform the following functions, using an external Distribution List residing on EDAPATH, or an HTML input form:

- Create a new Distribution List.
- Copy one Distribution List to another.
- Add new members to an existing Distribution List.
- Replace one or more members in an existing Distribution List.
- Delete a Distribution List.

Note: These functions cannot be performed as a pre-processing procedure of the schedule being submitted.

Syntax

How to Maintain Distribution Lists Using DSTBULK

```
-SET &var_name=DSTBULK(srv_userid,srv_userid_length,  
- 'srv_userpass',srv_userpass_length,  
- 'host_port',host_port_length,  
- 'name',name_length,  
- 'function',function_length,  
- 'access',access_length,  
- 'method',method_length,  
- 'user',user_length,  
- 'filename',filename_length,  
- 'copy',copy_length,  
- 'httpuser/pswd',httpuser/pswd_length,  
[- 'tcpiplevel',]  
- 'I4');
```

where:

&var_name

Is the variable that will contain the return code.

DSTBULK Subroutine Arguments

The following sections describe the arguments for the DSTBULK subroutine. For an example that uses the DSTBULK subroutine, see *Replacing Members in a Distribution List Using DSTBULK* on page 4-15.

srv_userid Argument for DSTBULK

Description	Valid user ID for the WebFOCUS Reporting Server. It must match the user ID stored in the ReportCaster Repository.
Required?	Yes
Format	A48 (Windows NT/2000 and UNIX) A8 (OS/390)
Valid Values	Single quotation marks, ampersands, and spaces are not allowed. First character must be alphanumeric (OS/390).

srv_userid_length Argument for DSTBULK

Description	Length of the user ID.
Required?	Yes
Format	I4
Valid Values	Integer

srv_userpass Argument for DSTBULK

Description	Valid password for the WebFOCUS Reporting Server user ID. It must match the password of the user ID stored in the ReportCaster Repository.
Required?	Yes
Format	A48
Valid Values	Alphanumeric characters and spaces are allowed.

srv_userpass_length Argument for DSTBULK

Description	Length of the WebFOCUS Reporting Server user password.
Required?	Yes
Format	I4
Valid Values	Integer

hostname_port Argument for DSTMEM

Description	<p>Is the host name (or IP address) and port number of the Web server on which the WebFOCUS Client is installed.</p> <ul style="list-style-type: none"> If the port number is 80 (the default), you may omit the colon and port number. For example, <i>'hostname'</i>, If the port number is not 80, use a colon as a delimiter and then specify the port number. For example, <i>'hostname:81'</i>,
Required?	Yes
Format	A64
Valid Values	No restrictions.

hostname_port_length Argument for DSTMEM

Description	Is the length of the host name and port number. This will be only the length of the host name if the port number is omitted.
Required?	Yes
Format	I4
Valid Values	Integer

name Argument for DSTBULK

Description	Name of the Distribution List that will be modified.
Required?	Yes
Format	A50
Valid Values	First character must be alphabetic.

name_length Argument for DSTBULK

Description	Length of the name of Distribution List.
Required?	Yes
Format	I4
Valid Values	Integer

function Argument for DSTBULK

Description	Type of maintenance action that will be performed.
Required?	Yes
Format	A1
Valid Values	C = Create a new Distribution List. N = Add new members to an existing Distribution List. R = Replace all members in an existing Distribution List. D = Delete an existing Distribution List.

function_length Argument for DSTBULK

Description	Length of the value of the function.
Required?	Yes
Format	I4
Valid Values	1

access Argument for DSTBULK

Description	Length of the value of the function.
Required?	Yes, if <i>function</i> is C. Ignored if <i>function</i> is N, R, or D.
Format	A2
Valid Values	PR = Private Distribution List (other users cannot access the list). PU = Public Distribution List (anyone can access the list).

access_length Argument for DSTBULK

Description	Length of the value of the access allowed.
Required?	Yes, if <i>access</i> is supplied.
Format	I4
Valid Values	2

method Argument for DSTBULK

Description	Distribution method for a new Distribution List.
Required?	Yes, if <i>function</i> is C. Ignored if <i>function</i> is N, R, or D.
Format	A5
Valid Values	FTP or F = FTP distribution. MAIL or M = E-mail distribution. PRINT or P = Printer distribution.

method_length Argument for DSTBULK

Description	Length of the value for the distribution method.
Required?	Yes, if <i>function</i> is C.
Format	I4
Valid Values	Integer

user Argument for DSTBULK

Description	One of the following: <ul style="list-style-type: none"> User ID of the owner of the target Distribution List when <i>copy</i> is not present. The target Distribution List is the one being created, added to, replaced, or deleted. User ID of the owner of the source Distribution List when <i>copy</i> is present. The owner of the target Distribution List is the WebFOCUS Reporting Server user ID, passed as the first argument to the subroutine.
Required?	Yes This feature enables a logon user to create or manipulate a Distribution List for another user, whose ID is supplied on <i>owner</i> .
Format	A48 (Windows NT/2000 and UNIX) A8 (OS/390)
Valid Values	Single quotation marks, ampersands, and spaces are not allowed. First character must be alphanumeric (OS/390).

user_length Argument for DSTBULK

Description	Length of the user argument.
Required?	Yes
Format	I4
Valid Values	Integer

filename Argument for DSTBULK

Description	<p>Name of the external source file used to supply member names. The following requirements apply:</p> <ul style="list-style-type: none"> • The file must be located in EDAPATH (Windows NT/2000 and UNIX), or in a partitioned data set (PDS) allocated to ddname EDARPC (OS/390). • The file must have the extension .fex on Windows NT/2000 and UNIX, although you do not specify it. <p>Note: This is similar to the way the IBIB_filename parameter is used for DSTDLBULK. For more information, see the section about creating an external file in Chapter 3, <i>ReportCaster Servlet API</i>.</p>
Required?	Yes, if supplying members from an external file when <i>function</i> is C, N, or R.
Format	A8
Valid Values	<p>First character must be alphanumeric. Single quotation marks and ampersands are not allowed. Blank lines are not allowed in an external distribution file.</p>

filename_length Argument for DSTBULK

Description	Length of the name of the external source file.
Required?	Yes, if <i>filename</i> is supplied.
Format	I4
Valid Values	I4

copy Argument for DSTBULK

Description	Name of the source Distribution List. Applies when <i>function</i> is one of the following: <ul style="list-style-type: none"> • C, which copies an existing Distribution List to a new Distribution List. • N, which appends an existing Distribution List to another one. • R, which replaces an existing Distribution List with another one.
Required?	Yes, if supplying members from a Distribution List.
Format	A50
Valid Values	First character must be alphabetic.

copy_length Argument for DSTBULK

Description	Length of the name of the source Distribution List.
Required?	Yes, if <i>copy</i> is supplied.
Format	I4
Valid Values	Integer

httpuser/pswd Argument for DSTBULK

Description	Web server user ID and password.
Required?	Yes, if the Web server is running with security on.
Format	A75
Valid Values	Valid Web server user ID and password.

httpuser/pswd_length Argument for DSTBULK

Description	Length of the Web server user ID and password, including the character <i>/</i> .
Required?	Yes, if <i>httpuser/pswd</i> is supplied.
Format	I4
Valid Values	Integer

tcpiplevel Argument for DSTBULK

Description	Method ReportCaster uses for securing sockets from TCP/IP, based on a specific SAS/C library.
Required?	Yes, if the WebFOCUS Reporting Server runs on OS/390.
Format	I4
Valid Values	1 = Resolves special connectivity problems. 0 = Specifies no special connectivity problems.

returncode Argument for DSTBULK

Description	Value that DSTMEM returns to the calling procedure, indicating successful completion (0) or an error condition. For more information, see Appendix B, <i>ReportCaster API Messages and Codes</i> .
Required?	Yes
Format	I4
Valid Values	'I4'

Example

Replacing Members in a Distribution List Using DSTBULK

The following Dialogue Manager procedure calls DSTBULK to replace members in an existing Distribution List with members from an external file. The procedure resides on a WebFOCUS Reporting Server for Windows NT/2000. The numbers to the left of the code refer to the annotations that follow.

```

1. FILEDEF SUBLOG DISK D:\ibi\srv436\wfs\catalog\sublog.ftm
   -RUN
2. -SET &SUBERR = DSTBULK(
   - 'userid',6,
3. - 'mypass',6,
4. - 'ibihost',7,
5. - 'Managers',8,
6. - 'R',1,
7. - 'PU',2,
8. - ' ',0,
9. - 'JXDMKT',6,
10. - 'extfile1',8,
11. - ' ',0,
12. - ' ',0,
13. - 'I4');
14. -IF &SUBERR EQ 0 GOTO EXIT;
   -INCLUDE DSTBLKER
   -WRITE SUBLOG &SUBERR
   -EXIT

```

The procedure runs as follows:

1. The FILEDEF command establishes the location of a file named sublog.ftm. DSTBULK will write error codes and messages (if there are any) to this file, as coded in step 14. The -RUN command executes the FILEDEF command.

For more information about the FILEDEF command, see your *WebFOCUS Developing Reporting Applications* manual.

2. The -SET Dialogue Manager command calls the ReportCaster API subroutine DSTBULK. It sets the amper variable &SUBERR to the value of the return code that is provided on successful, or non-successful, completion of the subroutine.

The WebFOCUS Reporting Server user ID ('userid'), included in single quotation marks, and length of 'userid' (6), are the first arguments passed to the subroutine.

3. The WebFOCUS Reporting Server password ('mypass') and length of the password (6) are passed to the subroutine.
4. The name of the WebFOCUS Reporting Server is 'ibihost' and the port number is 80. The value 80 is not specified because it is the default. The length of the server name is 7.
5. The name of the Distribution List that will be modified is 'Managers'. The length of the name is 8.
6. The members in the Distribution List 'Managers' will be replaced with other members, as specified by the value 'R'. In this case, the source of the members is an external file. The length of the value 'R' is 1.
7. When a new Distribution List is being created, the access allowed (public or private) must be specified. However, in this example, member names from an external file will replace member names in an existing Distribution List. Therefore, this value will be ignored even though it is coded. The length of the value 'PU' is 2.
8. This line illustrates the required code for the arguments *method* and *method_length*. The argument *method* is an optional alphanumeric argument; it is only required if you are creating a new Distribution List. If you do not supply a value for an optional alphanumeric argument, you must code the line as shown to hold the position for the argument and to indicate a length of zero.

9. The ID of the owner of the Distribution List is 'JXDMKT'. The length of that value is 6.
10. The name of the external file that is the source of member names is 'extfile1'. The length of the name is 8. An external file is typically created by third-party software.
11. This line illustrates the required code for the arguments *copy* and *copy_length*. The argument *copy* is an optional alphanumeric argument; it is only required if you are supplying member names from an existing Distribution List. If you do not supply a value for an optional alphanumeric argument, you must code the line as shown to hold the position for the argument and to indicate a length of zero.
12. This line contains the code for the arguments *httpuser/pswd* and *httpuser/pswd_length*. These arguments are required only if the Web server is running with security on. In our example, Web server security is off, so values for the arguments are not supplied. However, the line must be coded as shown to indicate the position of the argument and a length of zero.
13. The format of the value of the return code ('I4') is the last argument passed to the subroutine.
14. The error handling code tests the value of &SUBERR, which holds the return code from the subroutine. If the return code is 0, which indicates successful completion of the subroutine, the procedure terminates.

If it is a non-zero value, indicating an error condition, the procedure continues to the next line, which incorporates the supplied file DSTBLKER. DSTBLKER translates return codes into meaningful messages. Messages are written to the file sublog.ftm, which was defined by the FILEDEF command in the first line of code.

For more information about DSTBLKER, see Appendix B, *ReportCaster API Messages and Codes*.

Note: The initial hyphen and space on each line are required for a continuation line.

Maintaining Single Distribution List Members Using the DSTMEM Subroutine

The DSTMEM subroutine maintains single members in a Distribution List stored in BOTDEST. It is a procedure-based call. With this subroutine you can:

- Add a single member to an existing Distribution List.
- Delete a single member from an existing Distribution List.

Syntax

How to Maintain Single Distribution List Members Using DSTMEM

```
-SET &var_name=DSTEMEM(srv_userid,srv_userid_length,
- 'srv_userpass',srv_userpass_length,
- 'host_port',host_port_length,
- 'name',name_length,
- 'function',function_length,
- 'owner',user_length,
- 'fldvlu',fldvlu_length,
- 'destfn',destfn_length,
- 'httpuser/pswd',httpuser/pswd_length,
[- 'tcpiplevel',]
- 'I4');
```

where:

&var_name

Is the variable that will contain the return code.

DSTEMEM Subroutine Arguments

The following sections describe the arguments for the DSTMEM subroutine. For an example that uses the DSTMEM subroutine, see *Adding a New Member to a Distribution List Using DSTMEM* on page 4-24.

srv_userid Argument for DSTMEM

Description	Valid user ID for the WebFOCUS Reporting Server.
Required?	Yes
Format	A48 (Windows NT/2000 and UNIX) A8 (OS/390)
Valid Values	Single quotation marks, ampersands, and spaces are not allowed. First character must be alphanumeric (OS/390).

srv_userid_length Argument for DSTMEM

Description	Length of the user ID.
Required?	Yes
Format	I4
Valid Values	Integer

srv_userpass Argument for DSTMEM

Description	Valid password for the WebFOCUS Reporting Server user ID.
Required?	Yes
Format	A48
Valid Values	Alphanumeric characters and spaces are allowed.

srv_userpass_length Argument for DSTMEM

Description	Length of the WebFOCUS Reporting Server user password.
Required?	Yes
Format	I4
Valid Values	Integer

hostname_port Argument for DSTMEM

Description	<p>Is the host name (or IP address) and port number of the Web server on which the WebFOCUS Client is installed.</p> <ul style="list-style-type: none"> • If the port number is 80 (the default), you may omit the colon and port number. For example, <i>'hostname'</i>, • If the port number is not 80, use a colon as a delimiter and then specify the port number. For example, <i>'hostname:81'</i>,
Required?	Yes
Format	A64
Valid Values	No restrictions.

hostname_port_length Argument for DSTMEM

Description	Is the length of the host name and port number. This will only be the length of the host name if the port number is omitted.
Required?	Yes
Format	I4
Valid Values	Integer

name Argument for DSTMEM

Description	Name of the Distribution List that will be modified.
Required?	Yes
Format	A50
Valid Values	First character must be alphabetic.

name_length Argument for DSTMEM

Description	Length of the name of the Distribution List.
Required?	Yes
Format	I4
Valid Values	Integer

function Argument for DSTMEM

Description	Type of maintenance action that will be performed.
Required?	Yes
Format	A1
Valid Values	A = Add a new member. D = Delete an existing member.

function_length Argument for DSTMEM

Description	Length of the value of the function.
Required?	Yes
Format	I4
Valid Values	1

owner Argument for DSTMEM

Description	User ID of the owner of the Distribution List being added to or deleted from.
Required?	Yes
Format	A48 (Windows NT and UNIX) A8 (OS/390)
Valid Values	Single quotation marks, ampersands, and spaces are not allowed. First character must be alphanumeric.

user_length Argument for DSTMEM

Description	Length of the user ID of the owner of the Distribution List being added to or deleted from.
Required?	Yes
Format	I4
Valid Values	Integer

fldvlu Argument for DSTMEM

Description	Field value on which the report is burst.
Required?	Yes, if the report is burst.
Format	A75
Valid Values	Burst value

fldvlu_length Argument for DSTMEM

Description	Length of the burst value.
Required?	Yes, if <i>fldvlu</i> is supplied.
Format	I4
Valid Values	I4

destfn Argument for DSTMEM

Description	Destination of the report or report section.
Required?	Yes
Format	A75
Valid Values	File name if the method is FTP. E-mail address if the method is MAIL. Printer name if the method is PRINT.

destfn_length Argument for DSTMEM

Description	Length of destination value.
Required?	Yes
Format	I4
Valid Values	Integer

httpuser/pswd Argument for DSTMEM

Description	Web server user ID and password.
Required?	Yes, if the Web server is running with security on.
Format	A75
Valid Values	Valid Web server user ID and password.

httpuser/pswd_length Argument for DSTMEM

Description	Length of the Web server user ID and password, including the character <i>l</i> .
Required?	Yes, if <i>httpuser/pswd</i> is supplied.
Format	I4
Valid Values	Integer

tcpiplevel Argument for DSTMEM

Description	Method ReportCaster uses for securing sockets from TCP/IP, based on a specific SAS/C library.
Required?	Yes, if the WebFOCUS Reporting Server runs on OS/390.
Format	I4
Valid Values	1 = Resolves special connectivity problems. 0 = Specifies no special connectivity problems.

returncode Argument for DSTMEM

Description	Value that DSTMEM returns to the calling procedure, indicating successful completion (0) or an error condition. For more information, see Appendix B, <i>ReportCaster API Messages and Codes</i> .
Required?	Yes
Format	I4
Valid Values	'I4'

Example**Adding a New Member to a Distribution List Using DSTMEM**

The following Dialogue Manager procedure calls DSTMEM to add a new member to a Distribution List. The procedure resides on a WebFOCUS Reporting Server on Windows NT/2000. The numbers to the left of the code refer to the annotations that follow.

```
1. FILEDEF SUBLOG DISK D:\ibi\srv436\wfs\catalog\sublog.ftm
   -RUN
2. -SET &SUBERR = DSTMEM(
   - 'userid',6,
3. - 'mypass',6,
4. - 'ibiwnt:3794',11,
5. - 'Managers',8,
6. - 'A',1,
7. - 'listownerId',11,
8. - &VAR1,&VAR2,
9. - &VAR3,&VAR4,
10. - ' ',0,
11. - 'I4');
12. -IF &SUBERR EQ 0 GOTO EXIT;
   -INCLUDE DSTMEMER
   -WRITE SUBLOG &SUBERR
   -EXIT
```

The procedure runs as follows:

1. The FILEDEF command establishes the location of a file named sublog.ftm. DSTMEM will write error codes and messages (if there are any) to this file, as coded in step 12. The -RUN command executes the FILEDEF command.

For more information about the FILEDEF command, see your *WebFOCUS Developing Reporting Applications* manual.

2. The -SET Dialogue Manager command calls the ReportCaster API subroutine DSTMEM. It sets the amper variable &SUBERR to the value of the return code that is provided on successful, or non-successful, completion of the subroutine.

The WebFOCUS Reporting Server user ID ('userid'), included in single quotation marks, and length of 'userid' (6), are the first arguments passed to the subroutine.
3. The WebFOCUS Reporting Server password ('mypass') and length of the password (6) are passed to the subroutine.
4. The name of the WebFOCUS Reporting Server is ibiwnt and the port number is 3794. The length of this string is 11. The string is enclosed in single quotation marks.
5. The name of the Distribution List that will be modified is 'Managers'. The length of the name is 8.
6. A new member name will be added to the Distribution List, as specified by the value 'A'. The length of the value 'A' is 1.
7. The ID of the owner of the Distribution List is 'listownerId'. It is 11 characters in length.

8. This line illustrates the required code for the arguments *fldvlu* and *fldvlu_length*. The argument *fldvlu* is an optional alphanumeric argument; it is only required if the report is burst.
9. In this example, these amper variables are where you can enter the e-mail address (&VAR3) and the length of the e-mail address (&VAR4) to be added to the Distribution List. There are no single quotation marks around the amper variables.
10. This line contains the code for the arguments *httpuser/pswd* and *httpuser/pswd_length*. These arguments are required only if the Web server is running with security on. In our example, Web server security is off, so values for the arguments are not supplied. However, the line must be coded as shown to indicate the position of the argument and a length of zero.
11. The format of the value of the return code ('I4') is the last argument passed to the subroutine.
12. The error handling code tests the value of &SUBERR, which holds the return code from the subroutine. If the return code is 0, which indicates successful completion of the subroutine, the procedure terminates.

If it is a non-zero value, indicating an error condition, the procedure continues to the next line, which incorporates the supplied file DSTMEMER. DSTMEMER translates return codes into meaningful messages. Messages are written to the file sublog.ftm, which was defined by the FILEDEF command in the first line of code.

For more information about DSTMEMER, see Appendix B, *ReportCaster API Messages and Codes*.

Note: The initial hyphen and space on each line are required for a continuation line.

Running a Scheduled Job Using the DSTRUN Subroutine

The DSTRUN subroutine is a procedure-based call that immediately runs a job that has previously been scheduled. This only pertains to server-based schedules.

With this subroutine you can pass values to the scheduled job that you are running. The values are assigned to the appropriate variables in the scheduled job and used at run time.

The specified job will run as soon as a server agent is available. The status of the job (active or inactive) is ignored.

Reference

DSTRUN Requirements

- The Distribution List used for the job must already exist.
- The schedule information for the job must already exist. A ReportCaster Administrator can determine a schedule ID for a job by running the report from the ReportCaster Console. Both administrators and users can determine a schedule ID from the Log File icon on the ReportCaster user interface. This is accessible from the Developer Studio, Managed Reporting, and Dashboard environments.
- The call must be made from a secure environment. This means that, the user must log on to your application with a valid WebFOCUS Reporting Server user ID and password. The user ID for the server must match the user ID stored in the ReportCaster Repository with the target job schedule.
- To run another user's job, you must supply the user's ID for the *owner* argument.
- If there are multiple schedules for a report request (job), you must supply the argument *scheduleid*.
- If there is one schedule for a job, you may supply either *scheduleid* or *jobdesc*.

Syntax

How to Immediately Run a Scheduled Job Using DSTRUN

```
-SET &var_name=DSTRUN(srv_userid,srv_userid_length,  
- 'srv_userpass',srv_userpass_length,  
- 'hostname_port',hostname_port_length,  
- 'scheduleid',scheduleid_length,  
- 'jobdesc',jobdesc_length,  
- priority,  
- 'owner',user_length,  
- 'parm',parm_length,  
- 'httpuser/pswd',httpuser/pswd_length,  
[- 'tcpiplevel,']  
- 'I4');
```

where:

&var_name

Is the variable that will contain the return code.

DSTRUN Subroutine Arguments

The following sections describe the arguments for the DSTRUN subroutine. For an example that uses the DSTRUN subroutine, see *Immediately Running a Scheduled Job Using DSTRUN* on page 4-32.

srv_userid Argument for DSTRUN

Description	Valid user ID for the WebFOCUS Reporting Server.
Required?	Yes
Format	A48 (Windows NT/2000 and UNIX) A8 (OS/390)
Valid Values	Single quotation marks, ampersands, and spaces are not allowed. First character must be alphanumeric (OS/390).

srv_userid_length Argument for DSTRUN

Description	Length of the user ID.
Required?	Yes
Format	I4
Valid Values	Integer

srv_userpass Argument for DSTRUN

Description	Valid password for the WebFOCUS Reporting Server user ID.
Required?	Yes
Format	A48
Valid Values	Alphanumeric characters and spaces are allowed.

srv_userpass_length Argument for DSTRUN

Description	Length of the WebFOCUS Reporting Server user password. You must use a colon to separate the server name and port number.
Required?	Yes
Format	I4
Valid Values	Integer

hostname_port Argument for DSTRUN

Description	Is the host name (or IP address) and port number of the Web server on which the WebFOCUS Client is installed. <ul style="list-style-type: none"> If the port number is 80 (the default), you may omit the colon and port number. For example, <i>'hostname'</i>, If the port number is not 80, use a colon as a delimiter and then specify the port number. For example, <i>'hostname:81'</i>,
Required?	Yes
Format	A64
Valid Values	No restrictions.

hostname_port_length Argument for DSTRUN

Description	Is the length of the host name and port number. This will only be the length of the host name if the port number is omitted.
Required?	Yes
Format	I4
Valid Values	Integer

scheduleid Argument for DSTRUN

Description	Unique, API-generated key that identifies the scheduled job.
Required?	Yes, if there are multiple schedules for a job. If the job was scheduled to run once, either this argument or <i>jobdesc</i> is required.
Format	A12
Valid Values	First character must be alphanumeric. Single quotation marks and ampersands are not allowed.

scheduleid_length Argument for DSTRUN

Description	Length of <i>scheduleid</i> .
Required?	Yes, if <i>scheduleid</i> is supplied.
Format	I4
Valid Values	Integer

jobdesc Argument for DSTRUN

Description	Unique, user-supplied description for the scheduled job.
Required?	Yes, if the job was scheduled to run once and <i>scheduleid</i> is not supplied.
Format	A90
Valid Values	Alphanumeric characters with embedded blanks and special characters are allowed.

jobdesc_length Argument for DSTRUN

Description	Length of the description for the scheduled job.
Required?	Yes, if <i>jobdesc</i> is supplied.
Format	I4
Valid Values	Integer

priority Argument for DSTRUN

Description	Priority level for the job scheduled to run.
Required?	Yes
Format	I4
Valid Values	1 = Highest priority 2 = Class 2 priority 3 = Class 3 priority 4 = Class 4 priority 5 = Lowest priority

owner Argument for DSTRUN

Description	User ID of the owner of the scheduled job.
Required?	Yes. This feature enables a logon user to run a job owned by another user, whose ID is supplied on <i>owner</i> .
Format	A48 (Windows NT/2000 and UNIX) A8 (OS/390)
Valid Values	Single quotation marks, ampersands, and spaces are not allowed. First character must be alphanumeric (OS/390).

user_length Argument for DSTRUN

Description	Length of the user ID of the owner of the scheduled job.
Required?	Yes
Format	I4
Valid Values	Integer

parm Argument for DSTRUN

Description	Values passed to the scheduled job. The values are assigned to variables in the scheduled job and used at run time. Only one value is supported for each argument specified.
Required?	No
Format	A75
Valid Values	Must fit on an 80-byte line, which includes all punctuation.

parm_length Argument for DSTRUN

Description	Length of the parameter string.
Required?	Yes, if <i>parm</i> is supplied. This value is the total parameter length, meaning that it may include multiple parameters in a single string.
Format	I4
Valid Values	I4

httpuser/pswd Argument for DSTRUN

Description	Web server user ID and password.
Required?	Yes, if the Web server is running with security on.
Format	A75
Valid Values	Valid Web server user ID and password.

httpuser/pswd_length Argument for DSTRUN

Description	Length of the Web server user ID and password, including the character <i>l</i> .
Required?	Yes, if <i>httpuser/pswd</i> is supplied.
Format	I4
Valid Values	Integer

tcpiplevel Argument for DSTRUN

Description	Method ReportCaster uses for securing sockets from TCP/IP, based on a specific SAS/C library.
Required?	Yes, if the WebFOCUS Reporting Server runs on OS/390.
Format	I4
Valid Values	1 = Resolves special connectivity problems. 0 = Specifies no special connectivity problems.

returncode Argument for DSTRUN

Description	Value that DSTRUN returns to the calling procedure, indicating successful completion (0) or an error condition. For more information, see Appendix B, <i>ReportCaster API Messages and Codes</i> .
Required?	Yes
Format	I4
Valid Values	'I4'

Example **Immediately Running a Scheduled Job Using DSTRUN**

The following Dialogue Manager procedure calls DSTRUN to immediately run a scheduled job. The procedure resides on a WebFOCUS Reporting Server for Windows NT/2000. The numbers to the left of the code refer to the annotations that follow.

```
1. FILEDEF SUBLOG DISK D:\ibi\srv436\wfs\catalog\sublog.ftm
   -RUN
2. -SET &SUBERR = DSTRUN(
   - 'userid',6,
3. - 'mypass',6,
4. - 'webhost',7,
5. - ' ',0,
6. - 'myjob',5,
7. - 5,
8. - 'cdmsales',8,
9. - 'COUNTRY=ENGLAND,CAR=JAGUAR',25,
10. - 'webuser/webpass',15,
11. - 'I4');
12. -IF &SUBERR EQ 0 GOTO EXIT;
   -INCLUDE DSTRUNER
   -WRITE SUBLOG &SUBERR
   -EXIT
```

The procedure runs as follows:

1. The FILEDEF command establishes the location of a file named sublog.ftm. DSTRUN will write error codes and messages (if there are any) to this file, as coded in step 12. The -RUN command executes the FILEDEF command.

For more information about the FILEDEF command, see your *WebFOCUS Developing Reporting Applications* manual.

2. The -SET Dialogue Manager command calls the ReportCaster API subroutine DSTRUN. It sets the amper variable &SUBERR to the value of the return code that is provided on successful, or non-successful, completion of the subroutine.

The WebFOCUS Reporting Server user ID ('userid'), included in single quotation marks, and length of 'userid' (6), are the first arguments passed to the subroutine.

3. The WebFOCUS Reporting Server password ('mypass') and length of the password (6) are passed to the subroutine.
4. The name of the WebFOCUS Reporting Server is 'webhost' and the port number is 80. The value 80 is not specified because it is the default. The length of the server name is 7.

5. This line illustrates the required code for the arguments *scheduleid* and *scheduleid_length*. The argument *scheduleid* is an optional alphanumeric argument; it is only required if there are multiple schedules for a job. If you do not supply a value for an optional alphanumeric argument, you must code the line as shown to hold the position for the argument and to indicate a length of zero.
6. The user-supplied description of the job is 'myjob'. The length of the description is 5.
7. The priority level is the lowest possible, 5.
8. The ID of the user who owns the scheduled job is 'cdmsales'. The length of the ID is 8.
9. Two parameters are being passed. The value ENGLAND is assigned to the variable COUNTRY, and the value JAGUAR is assigned to the variable CAR. The variables exist in the report request and require values at run time. They are enclosed in single quotation marks. The total length of this string is 25.
10. In our example, the Web server is running with security on, so a valid Web server user ID and password are required. The Web server user ID (webuser) and password (webpass) are passed to the subroutine. The length of the user ID and password string is 15, which includes the character /. The string is enclosed in single quotation marks.
11. The format of the value of the return code ('I4') is the last argument passed to the subroutine.
12. The error handling code tests the value of &SUBERR, which holds the return code from the subroutine. If the return code is 0, which indicates successful completion of the subroutine, the procedure terminates.

If it is a non-zero value, indicating an error condition, the procedure continues to the next line, which incorporates the supplied file DSTRUNER. DSTRUNER translates return codes into meaningful messages. Messages are written to the file sublog.ftm, which was defined by the FILEDEF command in the first line of code.

For more information about DSTRUNER, see Appendix B, *ReportCaster API Messages and Codes*.

Note: The initial hyphen and space on each line are required for a continuation line.

Using Amper Variables Within a Subroutine

To use amper variables within a subroutine:

- An ampersand (&) must be included as the first character before the amper variable name.
- Single quotation marks cannot be placed around amper variables.
- A second amper variable to code the length of the value (in accordance with the format needed to execute a subroutine) must be used with each amper variable.

Example

Using Amper Variables for the Burst Value and E-mail Address Field

You can use amper variables in place of hard coding a value that is passed as a parameter to a subroutine. This example uses the -SET Dialogue Manager command to supply:

- Values to the *parm* argument.
- The length of the *parm* argument (*parm_length*).

The variables &VAR1 and &VAR2 are then substituted for these arguments.

```
FILEDEF SUBLOG DISK D:\ibi\srv436\wfs\catalog\sublog.ftm
-RUN
-SET &VAR1='COUNTRY=FRANCE';
-SET &VAR2=14;
-SET &SUBERR = DSTMEM(
- 'userid',6,
- 'mypass',6,
- 'ibiwnt:3794',11,
- 'Managers',8,
- 'A',1,
- 'listownerId',11,
- &VAR1,&VAR2,
- ' ',0,
- 'I4');
-IF &SUBERR EQ 0 GOTO EXIT;
-INCLUDE DSTMEMER
-WRITE SUBLOG &SUBERR
-EXIT
```

CHAPTER 5

ReportCaster API JSP Samples

Topics:

- Scheduling Functions
- Log Functions
- Logging on to the ReportCaster Bean API
- JSP Samples

A set of sample ReportCaster API JavaServer™ Pages (JSP) that use Bean functionality are distributed with the ReportCaster API in the `ibi_html\broker` subdirectory under the WebFOCUS installation directory. Use these samples or customize them for your application needs.

Scheduling Functions

The following table lists the sample JavaServer Pages for scheduling functions that are distributed with the ReportCaster Bean API. These pages all use the DSTSchedFactory Bean and perform authentication.

ReportCaster Bean API Sample JavaServer Pages	Description	Main Method	Go To Page
rcaster_newschedule	Input form for a new schedule. Calls rcaster_create.jsp.	getServerProcedureHandler Required parameter: None	5-7, 5-13
rcaster_create	Creates a new schedule.	getCreateScheduleHandler Required parameter: All required schedule properties	5-13
rcaster_runonce	Runs a schedule once then deletes it from BOTSCHED.	getRunOnceScheduleHandler Required parameter: IBIB_scheduleid	5-13
rcaster_list	Generates a list of schedules.	GetScheduleListHandler Required parameter: None	5-13
rcaster_property	Displays main properties of a schedule.	getSchedulePropertyHandler Required parameter: IBIB_scheduleid	5-14
rcaster_detail	Generates the details of a schedule.	getScheduleDataHandler Required parameter: IBIB_scheduleid	5-15
rcaster_copy	Copies an existing schedule.	GetScheduleDataHandler getCreateScheduleHandler Required parameter: All required schedule properties	5-15
rcaster_update	Updates schedule information.	getUpdateScheduleHandler Required parameter: IBIB_scheduleid	5-15
rcaster_delete	Deletes a schedule from BOTSCHED.	getDeleteScheduleHandler Required parameter: IBIB_scheduleid	5-16

ReportCaster Bean API Sample JavaServer Pages	Description	Main Method	Go To Page
rcaster_run	Runs a schedule based on frequency.	getRunScheduleHandler Required parameter: IBIB_scheduleid	5-16
rcaster_setrun	Runs a schedule based on set priority.	getScheduleDataHandler Required parameter: IBIB_scheduleid	5-16
rcaster_clist	Generates a list of schedules based on selected conditions.	getPeriodScheduleHandler Required parameter: rbschclist.htm input	5-23
rcaster_runnowlog	Generates a list for schedules run immediately.	GetProcessLogHandler Required parameter: IBIB_jobnumber	5-24

Log Functions

The following table lists the sample JavaServer Pages for log functions that are distributed with the ReportCaster Bean API. These pages all use the DSTLogFactory Bean and perform authentication.

ReportCaster Bean API Sample JavaServer Pages	Description	Main Method	Go To Page
rcaster_loglist	Generates a list of schedules in BOTLOG and BOTLOG2.	getLogListHandler Required parameter: IBIB_jobnumber	5-13, 5-23
rcaster_logplist	Generates a list of processes for a schedule ID.	getSelectionHandler Required parameter: IBIB_jobnumber	5-24
rcaster_logpdelete	Deletes one or all processes for a schedule ID.	getDeleteProcessLogHandler Required parameter: IBIB_jobnumber	5-24
rcaster_logdelete	Deletes a schedule from BOTLOG and BOTLOG2.	getDeleteLogHandler Required parameter: IBIB_jobnumber	5-25

Procedure

How to Change the Agent Node and Agent Port Values

If you are not using the default value of localhost:8200 for the Agent Node and Agent Port to connect to the Distribution Server, you must manually change any sample pages that contain these values. The default value for agentProperty displays as follows:

```
<jsp:setProperty name = "auth" property="agentProperty" value="localhost:8200" />
```

- If you are using WebFOCUS Version 4 Release 3.6 Java Gen 251 and earlier, edit the following sample pages and manually change all default Agent Node and Agent Port values to the appropriate host name and port for your environment.

rcaster_clist.jsp	rcaster_logplist.jsp
rcaster_copy.jsp	rcaster_newschedule.jsp
rcaster_create.jsp	rcaster_property.jsp
rcaster_delete.jsp	rcaster_run.jsp
rcaster_detail.jsp	rcaster_runnowlog.jsp
rcaster_list.jsp	rcaster_runonce.jsp
rcaster_logdelete.jsp	rcaster_setrun.jsp
rcaster_loglist.jsp	rcaster_update.jsp
rcaster_logpdelete.jsp	

- If you are using WebFOCUS Version 4 Release 3.6 Service Pack 2, edit *drive:\ibi\WebFOCUS436\ibi_html\broker\rcaster_distport.jsp* and change the default value from localhost:8200 to the appropriate host name and port for your environment.

Logging on to the ReportCaster Bean API

The ReportCaster API requires a valid WebFOCUS logon. A sample logon form is distributed with ReportCaster in the following location

```
drive:\ibi\WebFOCUSrel\ibi_html\broker\rbalogon.html
```

where:

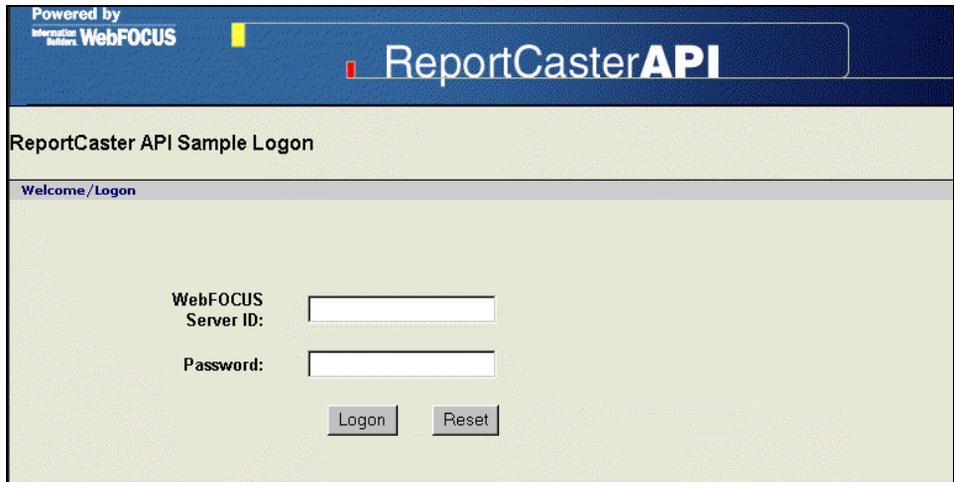
drive

Is the letter of the drive on which WebFOCUS is installed.

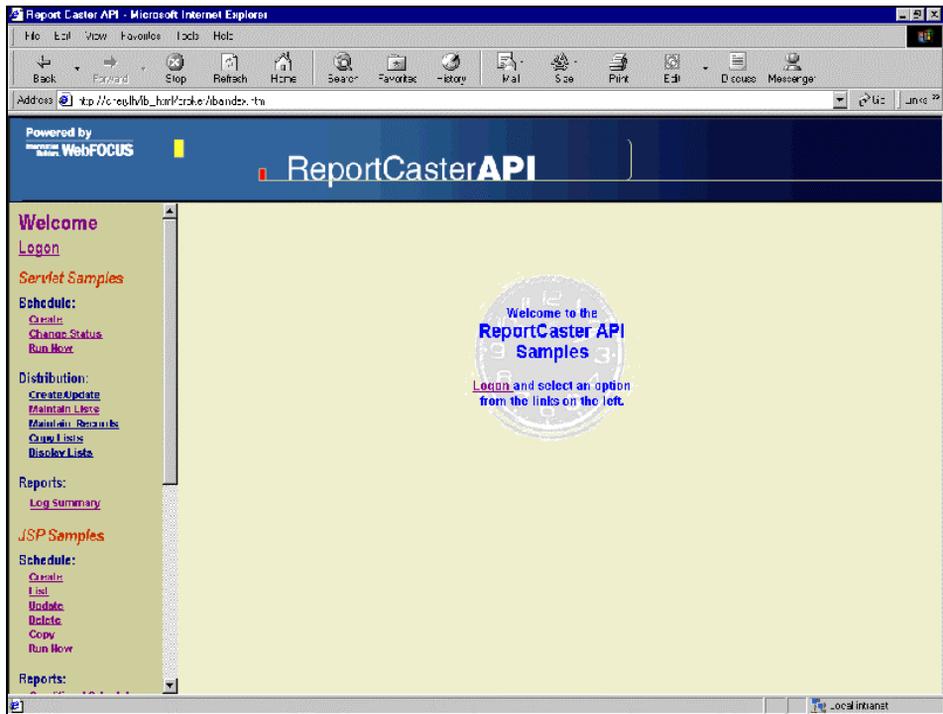
rel

Is the WebFOCUS release.

The ReportCaster API Sample Logon screen displays as follows:



After you enter a valid WebFOCUS Server ID and password, the ReportCaster API Welcome page displays:



The Welcome page contains links to:

- Servlet Samples. For more information, see Chapter 6, *ReportCaster API Servlet Samples*.
- JSP Samples. For more information, see *JSP Samples* on page 5-6.
- ReportCaster Repository Reports using a WebFOCUS servlet or a WebFOCUS CGI call. For more information, see Chapter 7, *ReportCaster Repository Reports*.

JSP Samples

Once you have logged into the ReportCaster API, you can click on the following JSP Samples on the left side of the API window:



JSP Schedule Options

This section describes how to create a schedule and the ways in which you can list, update, delete, copy, and run the schedule once it has been created.

Procedure

How to Create a Schedule Using a JSP

1. Click **Create** (runs `rcaster_newschedule.jsp`) in the Schedule section of the JSP Samples. The Create a New Schedule window displays:

On HP-UX 11 using the iPlanet Web server, parameters are lost when a page forward is performed within a JSP. Therefore, a null message displays when running the `rcaster_newschedule.jsp`. As a workaround, modify the `rcaster_newschedule.jsp` and replace the page forward with the actual code from the forwarded file.

2. Enter the following values in the Description section:
 - a. **Job name** (required). The name of the report request to be scheduled for execution and distribution. This must be specified in EDAPATH, and it can only be selected from the drop-down list.

Do not include the filename extension. The first character of this value must be alphanumeric. Single quotes and ampersands are not allowed.

Note: When creating a schedule using the `rcaster_newschedule.jsp` with a FOCUS Repository, do not use parentheses in the job name field.
 - b. **Job description** (optional). A unique, user-supplied description for the report request (job) being scheduled. Alphanumeric characters with embedded blanks and special characters are allowed.
 - c. **Pre-processing** (optional). The name of a procedure and its parameter string that will run prior to the scheduled report request. This procedure typically performs setup tasks.
 - d. **Post-processing** (optional). The name of a procedure and its parameter string that will run after the scheduled report request.
 - e. **Active** (required). The status of a report request. Valid values are Yes (Active) and No (Inactive).

3. Scroll to the Frequency section and enter the frequency parameters:

Frequency	
Interval:	<input type="radio"/> Once&Delete <input type="radio"/> Once <input type="radio"/> Hourly <input checked="" type="radio"/> Daily <input type="radio"/> Weekly <input type="radio"/> Monthly <input type="radio"/> Yearly
# times within Interval:	<input type="text" value="1"/>
Weekdays: <small>Required if frequency is weekly</small>	<input type="checkbox"/> Mon <input type="checkbox"/> Tue <input type="checkbox"/> Wed <input type="checkbox"/> Thu <input type="checkbox"/> Fri <input type="checkbox"/> Sat <input type="checkbox"/> Sun
Days of the month: <small>Required if frequency is monthly</small>	<input type="text" value="1"/>
Start date (YYYYMMDD):	<input type="text" value="20011010"/>
End date (YYYYMMDD):	<input type="text" value="20991231"/>
Start time (HHMM):	<input type="text" value="1611"/>
End time (HHMM):	<input type="text" value="2359"/>
Send to	
Address Type:	<input type="radio"/> Distribution List <input type="radio"/> Distribution Filename <input checked="" type="radio"/> Single Dest. (Email Address, FTP Filename, F
Distribution address:	<input type="text" value="chuck_hill@ibi.com"/>
Burst report:	<input type="radio"/> Yes <input checked="" type="radio"/> No

- a. **Interval** (required). The period of time (the interval) on which report execution and distribution are based. Valid values are:
 - Once&Delete—Runs the report and then deletes it from BOTSCHED.
 - Once.
 - Hourly.
 - Daily.
 - Weekly.
 - Monthly.
 - Yearly.
- b. **# times within interval** (optional). The number of times you want to run your report within the interval specified.
- c. **Weekdays** (optional). If you specified a weekly interval, this parameter is the day of the week the report request will run.

This parameter can occur multiple times. For example, if you specify a weekly interval, and specify 3 for the # of times within interval, the report runs every 3 weeks. Values are required for the specific days of the week such as Monday, Wednesday, and Friday.

- d. **Days of the month** (optional). If you specified a monthly interval, this parameter is the day of the month the report request will run.

This parameter can occur multiple times. For example, if you specify a monthly interval, and specify 2 for the # of times within interval parameter, the report runs every 2 months. Values are required for the specific days in the month, such as the 1st and the 15th.

- e. **Start date (YYYYMMDD)** (optional). The date of first report execution and distribution, in the format *YYYYMMDD*, where *YYYY* is the 4-digit year, *MM* is the month, and *DD* is the day of the month. The default is the current system date.
- f. **End date (YYYYMMDD)** (optional). The date of last report execution and distribution, in the format *YYYYMMDD*, where *YYYY* is the 4-digit year, *MM* is the month, and *DD* is the day of the month. The default is 20990101.
- g. **Start time (HHMM)** (optional). The time of first report execution and distribution, in the format *HHMM*, where *HH* is the hour and *MM* is the minute. The default is the current system time.
- h. **End time (HHMM)** (optional). The time of last report execution and distribution, in the format *HHMM*, where *HH* is the hour and *MM* is the minute. The default is 0000 (midnight).

4. Scroll to the Send to section and enter the send to parameters:

Send to	
Address Type:	<input type="radio"/> Distribution List <input type="radio"/> Distribution Filename <input checked="" type="radio"/> Single Dest. (Email Address, FTP Filename, F
Distribution address:	<input type="text" value="chuck_hill@ibi.com"/>
Burst report:	<input type="radio"/> Yes <input checked="" type="radio"/> No

- a. **Address Type** (required). The distribution method:
- Distribution List.
 - Distribution Filename. You must enter the fully qualified path of the file (for example, c:\ibi\apps\filename.fex). This must reside on EDAPATH.
 - Single Destination (single e-mail address, single FTP file name, or a single printer).
- b. **Distribution address** (required). A particular Distribution List, distribution file name, single e-mail address, FTP file name, or printer. For example, chuck_hill@ibi.com is a single e-mail address.
- c. **Burst report**. Possible values are Yes (burst) and No (no burst).

5. Scroll to the Send method section and enter the send method parameters:

- a. **Send via** (required). The distribution method. Possible values are e-mail, FTP, or printer.
- b. **Send format** (required). The format of your distributed report output. Possible values are:
 - WP for plain text.
 - HTML for hypertext (recipient must have a browser in order to view).
 - PS for PostScript (recipient must have a GhostView application in order to view).
 - DOC for plain text with ASCII page breaks.
 - PDF for Adobe Acrobat PDF format (recipient must have Adobe in order to view).
 - EXCEL (recipient must have EXCEL in order to view).
 - EXL2K (recipient must have EXCEL 2000 in order to view).
 - WK1, a Lotus 1-2-3 spreadsheet format.

Note:

- If you specified a distribution method of e-mail, proceed to step 6.
- If you specified a distribution method of FTP, proceed to step 7.
- If you specified a distribution method of printer, the output is sent to the printer selected in the Distribution List. No parameters are required. Proceed to step 8.

6. If you specified a distribution method of e-mail, enter the following e-mail parameters:
 - a. **Mail server** (required). The name of the mail server that distributes the report. Single quotes and ampersands are not allowed. The default value is the value specified in the ReportCaster configuration file (bkrsched.cfg).
 - b. **Reply Address** (required). The e-mail address for return mail (the reply address). Single quotes and ampersands are not allowed, and an @ character must be included.
 - c. **Subject** (optional). The content or purpose of the e-mail.
 - d. **Company** (optional). The company that the sender is associated with.
 - e. **Send As Attachment** (required). Specifies whether to send the e-mail as an attachment (Yes) or within the body of the e-mail (No).
7. If you specified a distribution method of FTP, enter the following FTP parameters:
 - a. **FTP server** (required). The name of the FTP server that distributes the report. Single quotes and ampersands are not allowed. The default value is the value specified in the ReportCaster configuration file (bkrsched.cfg).
 - b. **Location** (required). The destination of the FTP-distributed report. It must be a subdirectory of the FTP server root directory, or an alias defined to the FTP server. Single quotes and ampersands are not allowed.
 - c. **User** (required). The user ID authorized for FTP transfer. Single quotes and ampersands are not allowed. The default value is the value specified in the ReportCaster configuration file (bkrsched.cfg).
 - d. **Password** (required). The password associated with the user ID authorized for FTP transfer. Single quotes and ampersands are not allowed. The default value is the value specified in the ReportCaster configuration file (bkrsched.cfg).
 - e. **Index file name** (optional). The name of the FTP index file for a bursted report. The first character must be alphanumeric, and the default value is HOLD.

8. Scroll to the Notification section and enter the notification parameters:

Notification	
<input checked="" type="checkbox"/> Notify:	<input checked="" type="radio"/> Inactive <input type="radio"/> Always Notify <input type="radio"/> Notify of Error
<input checked="" type="checkbox"/> Notify Subject:	<input type="text" value="Turnover Ratio by Branch"/>
<input checked="" type="checkbox"/> Notify Address:	<input type="text" value="chuck_hill@ibi.com"/>
<input checked="" type="checkbox"/> Notify Device Info:	<input type="text"/>
<input checked="" type="checkbox"/> Notify Reply Address:	<input type="text" value="chuck_hill@ibi.com"/>

- a. **Notify** (required). The notification of job status. Possible values are:
 - Inactive (no notification is sent).
 - Always Notify (notification is always sent).
 - Notify of Error (content of the log is sent on an error condition).
 - b. **Notify Subject** (optional). The content or purpose of the notification message. Single quotes and ampersands are not allowed.
 - c. **Notify Address**. Required if Notify is set to Always Notify or Notify of Error. It is the e-mail address of the person who will receive full notification (complete log report).
 - d. **Notify Device Info** (optional). For a device other than e-mail.
 - e. **Notify Reply Address**. Required if Notify is set to Always Notify or Notify of Error. It is the e-mail address for return mail (the response to the notification message). Single quotes and ampersands are not allowed, and an @ character must be included.
9. Scroll to the Parameters section and enter the optional parameter names and values to pass to the scheduled job for each parameter name. The values are assigned to variables in the scheduled job and used at run time.

Parameters	
<input checked="" type="checkbox"/> Parameter Name	<input checked="" type="checkbox"/> Parameter Value
<input type="text" value="PLANT"/>	<input type="text" value="BOS"/>
<input type="text" value="PLANT"/>	<input type="text" value="LA"/>
<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>

10. Click **Create** (runs `rcaster_create.jsp`) or **Run Now** (runs `rcaster_runonce.jsp`). The Schedule List displays as follows:

Schedule List				
Please select schedule and click the button.				
List Schedule		Create Schedule		Show Log
				Total Schedule(s) 3
Properties	Detail	Copy	Delete	Run Now Set&Run...
		Job Name	Job Description	User
<input type="radio"/>		centhrv	Turnover Ratio by Branch	digcth
<input type="radio"/>		centordv	Product Orders Shipped by Manufacturing Plant	digcth
<input type="radio"/>		centqav	Quality Assurance Defect Ratio for Manufacturing Plants	digcth
Properties	Detail	Copy	Delete	Run Now Set&Run...
List Schedule		Create Schedule		Show Log

Note:

- When you click **Create**, the report job runs and an entry is created in BOTSCHEDED and BOTLOG.
- When you click **Run Now**, the report job runs and an entry is created only in BOTLOG.

Schedule List

You can view schedules that you have created by clicking **List** (runs `rcaster_list.jsp`) in the Schedule section of the JSP Samples. The Schedule List window displays:

Schedule List				
Please select schedule and click the button.				
List Schedule		Create Schedule		Show Log
				Total Schedule(s) 3
Properties	Detail	Copy	Delete	Run Now Set&Run...
		Job Name	Job Description	User
<input type="radio"/>		centhrv	Turnover Ratio by Branch	digcth
<input type="radio"/>		centordv	Product Orders Shipped by Manufacturing Plant	digcth
<input type="radio"/>		centqav	Quality Assurance Defect Ratio for Manufacturing Plants	digcth
Properties	Detail	Copy	Delete	Run Now Set&Run...
List Schedule		Create Schedule		Show Log

Note: You can also access the Schedule List window by clicking the **Update**, **Delete**, **Copy**, or **Run Now** links in the Schedule section of the JSP Samples.

The following options are available in the Schedule List window. You can:

- Use the radio button to select a specific schedule. Then click the desired action button (Properties, Detail, Copy, Delete, Run Now, Set&Run).
- Click **Create Schedule** (runs `rcaster_newschedule.jsp`) to create a schedule.
- Click **Show Log** (runs `rcaster_loglist.jsp`) to run the log summary report for a report you have selected (for example, Turnover Ratio by Branch).

- Click Properties (runs rcaster_property.jsp) to view the properties of a report you have selected (for example, Turnover Ratio by Branch). The Schedule Property window displays:

Schedule Property

Please click the button.

List Schedule	Create Schedule	Show Log			
Schedule: Turnover Ratio by Branch					
Detail	Copy	Delete	Run Now	Set&Run...	Reload
Schedule ID	St5s4I4i24				
Job Name	centhrv				
Job Description	Turnover Ratio by Branch				
Active	<input checked="" type="checkbox"/> YES				
User Name	digcth				
Next Run Time	Thu, Oct 11, 2001 04:11 PM EDT				
Focexec Origin	Server				
Send Method	 Email				
Address Type	Single File				
Address	chuck_hill@ibi.com				
Priority	3				
By Field	N				
Send Format	HTML				
Interval Type	Day				
List Schedule	Create Schedule	Show Log			

- Click **Detail** (runs rcaster_detail.jsp) to view the schedule details of a report you have selected (for example, Turnover Ratio by Branch). The Schedule Detail window displays:

Schedule Detail

Required fields are in **bold**. Place mouse over  for help on each field.

Schedule ID St5s4l4i24

Description

 **Job name:**

 Job description:

 Pre-processing:

 Post-processing:

 Active: Yes No

On the Schedule Detail window, you can:

- Make changes to your report request and click **Update** (runs rcaster_update.jsp) to reflect these changes.
- Click **Reload** (runs rcaster_detail.jsp). This is useful if you have made incorrect changes and would like to refresh your schedule detail window so that it displays the original data.
- Click **Copy** (runs rcaster_copy.jsp) to copy a report you have selected (for example, Turnover Ratio by Branch). The Schedule List window displays with the additional copied report:

Schedule List

Please select schedule and click the button.

Total Schedule(s) 4

		Job Name	Job Description	User
<input type="radio"/>		centhrv	(St5s8t90610) Copy of Turnover Ratio by Branch	digcth
<input type="radio"/>		centhrv	Turnover Ratio by Branch	digcth
<input type="radio"/>		centordv	Product Orders Shipped by Manufacturing Plant	digcth
<input type="radio"/>		centqav	Quality Assurance Defect Ratio for Manufacturing Plants	digcth

- Click Delete (runs rcaster_delete.jsp) to delete the selected report (for example, Copy of Turnover Ratio by Branch). The Schedule List window displays without the selected report:

Schedule List					
Please select schedule and click the button.					
List Schedule		Create Schedule		Show Log	
				Total Schedule(s) 3	
Properties	Detail	Copy	Delete	Run Now Set&Run...	
Job Name	Job Description			User	
<input type="checkbox"/> <input type="checkbox"/>	centhrv	Turnover Ratio by Branch			digcth
<input type="checkbox"/> <input type="checkbox"/>	centordv	Product Orders Shipped by Manufacturing Plant			digcth
<input type="checkbox"/> <input type="checkbox"/>	centqav	Quality Assurance Defect Ratio for Manufacturing Plants			digcth
Properties	Detail	Copy	Delete	Run Now Set&Run...	
List Schedule		Create Schedule		Show Log	

- Click Run Now (runs rcaster_run.jsp) to run the selected report (for example, Turnover Ratio by Branch). This creates an entry in the BOTLOG table.
- Click Set&Run (runs rcaster_setrun.jsp). The RunNow Schedule window displays:

RunNow Schedule	
Required fields are in bold . Place mouse over  for help on each field.	
List Schedule Create Schedule Show Log	
Schedule ID St5s4I4I24	
Properties	Detail Copy Delete Reload
Priority	
<input checked="" type="checkbox"/> Priority:	<input checked="" type="radio"/> 1 <input type="radio"/> 2 <input type="radio"/> 3 <input type="radio"/> 4 <input type="radio"/> 5
Parameters	
<input checked="" type="checkbox"/> PLANT	<input type="text" value="BOS"/>
<input type="button" value="Run"/>	
List Schedule Create Schedule Show Log	

You can then select the Priority of the schedule (1 is the highest and 5 is the lowest), enter optional parameters, and click Run (runs rcaster_run.jsp) to run the report.

JSP Reports

This section describes how to run the following reports:

- **Conditional Schedules.** This report allows you to get a list of schedules for a specific set of conditions. The schedules must have a ‘next run time’ greater than the current time.
- **Log Summary.** This report provides a list of schedules that have run.

Procedure**How to Run a Conditional Schedules Report**

1. Click Conditional Schedules (runs rbaschclist.htm) in the Reports section of the JSP Samples. The Display Schedules with Conditions window displays:

Display Schedules with Conditions

This report may be used for schedules that have a 'next run time'. Conditions may be specified below in order to limit your list result. Place mouse over  for help on each field.

Time Period	
 Period Start date (YYYYMMDD):	20011010
 Period End date (YYYYMMDD):	20991231
 Period Start time (HHMM):	1611
 Period End time (HHMM):	2359

Important:

You can enter any combination of the following optional parameters to produce your desired report. If you do not specify any parameters, all schedules that have a 'next run time' will display. All text fields are case sensitive, and must be entered in the same exact case as when the schedule was created. In this procedure, parameters have been entered for the Turnover Ratio by Branch schedule.

2. Enter the following optional time period parameters:
 - a. Period Start date (YYYYMMDD).** The date of first report execution and distribution, in the format *YYYYMMDD*, where *YYYY* is the 4-digit year, *MM* is the month, and *DD* is the day of the month. The report will display schedules with a 'next run time' on or after the date specified.
 - b. Period End date (YYYYMMDD).** The date of last report execution and distribution, in the format *YYYYMMDD*, where *YYYY* is the 4-digit year, *MM* is the month, and *DD* is the day of the month. The report will display schedules with a 'next run time' on or before the date specified.
 - c. Period Start time (HHMM).** The time of first report execution and distribution, in the format *HHMM*, where *HH* is the hour and *MM* is the minute. The report will display schedules with a 'next run time' on or after the time specified.
 - d. Period End time (HHMM).** The time of last report execution and distribution, in the format *HHMM*, where *HH* is the hour and *MM* is the minute. The report will display schedules with a 'next run time' on or before the time specified.

3. Scroll to the Description section and enter the description parameters:

Description	
Job name:	<input type="text" value="centhrv"/>
Job description:	<input type="text" value="Turnover Ratio by Branch"/>
Pre-processing:	<input type="text"/>
Post-processing:	<input type="text"/>
Active:	<input type="radio"/> No Requirement <input checked="" type="radio"/> Yes <input type="radio"/> No

- a. **Job name.** The name of the report request.
- b. **Job description.** A unique, user-supplied description for the report request (job).
- c. **Pre-processing.** The name of a procedure and its parameter string that ran prior to the scheduled report request.
- d. **Post-processing.** The name of a procedure and its parameter string that ran after the scheduled report request.
- e. **Active.** Denotes the status of a report request. Possible values are:
 - No Requirement (Active or Inactive).
 - Yes (Active).
 - No (Inactive).

4. Scroll to the Frequency section and enter the frequency parameters:

Frequency	
Interval:	<input checked="" type="radio"/> All Types <input type="radio"/> Once <input type="radio"/> Hourly <input type="radio"/> Daily <input type="radio"/> Weekly <input type="radio"/> Monthly <input type="radio"/> Yearly
# times within Interval:	<input type="text"/>
Weekdays: <small>Required if frequency is weekly</small>	<input type="checkbox"/> Mon <input type="checkbox"/> Tue <input type="checkbox"/> Wed <input type="checkbox"/> Thu <input type="checkbox"/> Fri <input type="checkbox"/> Sat <input type="checkbox"/> Sun
Days of the month: <small>Required if frequency is monthly</small>	<input type="text" value="1"/> <input type="text" value="2"/> <input type="text" value="3"/> <input type="text" value="4"/>
Start date (YYYYMMDD):	<input type="text" value="20011010"/>
End date (YYYYMMDD):	<input type="text" value="20091231"/>
Start time (HHMM):	<input type="text" value="1611"/>
End time (HHMM):	<input type="text" value="2359"/>

- a. **Interval.** The period of time (the interval) on which report execution and distribution are based. Possible values are:
 - All Types
 - Once
 - Hourly
 - Daily
 - Weekly
 - Monthly
 - Yearly
- b. **# times within interval.** The number of times you want to run your report within the interval specified.
- c. **Weekdays.** The day of the week the report request will run.
- d. **Days of the month.** The day of the month the report request will run.
- e. **Start date (YYYYMMDD).** The date of first report execution and distribution, in the format *YYYYMMDD*, where *YYYY* is the 4-digit year, *MM* is the month, and *DD* is the day of the month.
- f. **End date (YYYYMMDD).** The date of last report execution and distribution, in the format *YYYYMMDD*, where *YYYY* is the 4-digit year, *MM* is the month, and *DD* is the day of the month.
- g. **Start time (HHMM).** The time of first report execution and distribution, in the format *HHMM*, where *HH* is the hour and *MM* is the minute.
- h. **End time (HHMM).** The time of last report execution and distribution, in the format *HHMM*, where *HH* is the hour and *MM* is the minute.

5. Scroll to the Send to section and enter the send to parameters:

Send to

Address Type: All Address Types Distribution List Distribution Filename
 Single Dest. (Email Address, FTP Filename, Printer)

Distribution address:

Burst report: No Requirement Yes No

- a. Address Type.** Select one of the following distribution method options:
- All Address Types.
 - Distribution List.
 - Distribution Filename.
 - Single Destination (single e-mail address, single FTP file name, or a single printer).
- b. Distribution address.** You can specify a particular Distribution List, distribution file name, single e-mail address, FTP file name, or printer. For example, chuck_hill@ibi.com is a single e-mail address.
- c. Burst Report.** Possible values are:
- No Requirement (burst and no burst).
 - Yes (burst).
 - No (no burst).

6. Enter the following Send method parameters:

Send method

Send via: Email FTP Printer

Send format: HTML

E-mail:
Required if distribution method is via e-mail

Mail server: smtpserver

Reply Address: chuck_hill@ibi.com

Subject: _____

Company: _____

Send As Attachment: No Requirement Yes No

FTP:
Required if distribution method is via FTP

FTP server: _____

Location: _____

User: _____

Password: _____

Index file name: _____

Printer:
Output sent to printers as specified in the distribution list. No parameters required.

- a. **Send via.** The distribution method. Possible values are e-mail, FTP, or to a printer.
- b. **Send format.** The format of your report. Possible values are:
 - WP for plain text.
 - HTML for hypertext (recipient must have a browser in order to view).
 - PS for PostScript (recipient must have a GhostView application in order to view).
 - DOC for plain text with ASCII page breaks.
 - PDF for Adobe Acrobat PDF format (recipient must have Adobe in order to view).
 - EXCEL (recipient must have EXCEL in order to view).
 - EXL2K (recipient must have EXCEL 2000 in order to view).
 - WK1, a Lotus 1-2-3 spreadsheet format.

7. Enter the following e-mail parameters:

- a. **Mail server.** The mail server that distributes the report.
- b. **Reply Address.** The e-mail address for return mail (the reply address).
- c. **Subject.** The content or purpose of the e-mail.
- d. **Company.** The company that the sender is associated with.
- e. **Send As Attachment.** Specifies whether to send the e-mail with no requirements (No Requirement), as an attachment (Yes), or within the body of the e-mail (No).

8. Enter the following FTP parameters:
 - a. **FTP server.** The name of the FTP server that distributes the report.
 - b. **Location.** The destination of the FTP-distributed report.
 - c. **User.** The user ID authorized for FTP transfer.
 - d. **Password.** The password associated with the user ID.
 - e. **Index file name.** The name of the FTP index file for a bursted report.
9. Scroll to the Notification section and enter the notification parameters:

Notification

Notify: No Requirement Inactive Notify of Error Always Notify

Notify Subject:

Notify Address:

Notify Device Info:

Notify Reply Address:

Submit Reset

- a. **Notify.** Notification of job status. Possible values are:
 - No Requirement (includes Inactive, Notify of Error, and Always Notify).
 - Inactive (no notification is sent).
 - Notify of Error (content of the log is sent on an error condition).
 - Always Notify (notification is always sent).
- b. **Notify Subject.** The content or purpose of the notification message.
- c. **Notify Address.** The e-mail address of the person who will receive full notification (complete log report).
- d. **Notify Device Info.** For a device other than e-mail.
- e. **Notify Reply Address.** The e-mail address for return mail (the response to the notification message).

- Click Submit (runs rcaster_clist.jsp) to run the report. The Schedule List displays as follows:

Schedule List				
Please select schedule and click the button.				
List Schedule		Create Schedule		Show Log
Total Schedule(s) 3				
Properties	Detail	Copy	Delete	Run Now Set&Run...
Job Name	Job Description			User
<input type="radio"/>  centhrv	Turnover Ratio by Branch			digcth
<input type="radio"/>  centordv	Product Orders Shipped by Manufacturing Plant			digcth
<input type="radio"/>  centqav	Quality Assurance Defect Ratio for Manufacturing Plants			digcth
Properties	Detail	Copy	Delete	Run Now Set&Run...
List Schedule		Create Schedule		Show Log

Procedure

How to Run a Log Summary Report

- Click Log Summary (runs rcaster_loglist.jsp) in the Reports section of the JSP Samples. The Log Summary window displays:

Log Summary			
Please select schedule and click the button.			
List Schedule		Create Schedule	
		Show Log	
Log Summary 2			
Processes	Delete	Delete All	
Schedule ID	Job Name	Job Description	
<input type="radio"/> St5s4l4i24	centhrv	Turnover Ratio by Branch	
<input type="radio"/> St5s7cu386	centqav	Quality Assurance Defect Ratio for Manufacturing Plants	
Processes	Delete	Delete All	
List Schedule		Create Schedule	
		Show Log	

2. To view the processes of a schedule, select the schedule (for example, Turnover Ratio by Branch) and click Processes (runs rcaster_logplist.jsp). The Processes in Log window displays the Process ID, Start Time, and End Time of the selected schedule:

Processes in Log

Please select schedule and click the button.

List Schedule	Create Schedule	Show Log	
			Job Description Turnover Ratio by Branch
<input type="button" value="Show"/> <input type="button" value="Delete"/> <input type="button" value="Delete All"/>			
Process ID	Start Time	End Time	
<input type="radio"/> P0t5s4m9jq4	Wed, Oct 10, 2001 04:26:11 EDT	Wed, Oct 10, 2001 04:26:41	
<input type="radio"/> P0t5u6hk1pc	Thu, Oct 11, 2001 11:37:03 EDT	Thu, Oct 11, 2001 11:37:27 E	
<input type="button" value="Show"/> <input type="button" value="Delete"/> <input type="button" value="Delete All"/>			
List Schedule	Create Schedule	Show Log	

- a. Click Show (runs rcaster_runnowlog.jsp) to display the Schedule Runnow Log window for a selected Process ID (for example, P0t5u6hk1pc):

Schedule Runnow Log

Schedule runnow log.

List Schedule	Create Schedule	Show Log	
			Job Description: Turnover Ratio by Branch
<input type="button" value="Reload"/>			
Job Process Log Report			
Job Description: Turnover Ratio by Branch			
Server User: digcgh	(BTP1010) Starting worker thread		
Process: P0t5u6hk1pc	(BTP1010) Method: Mail. Host: ibismtp.		
Procedure: centhry	(BTP1010) Log report notification successfully sent to chuck_hill@ibi.com		
Schedule ID: St5s4l4l24			
Start Time: 2001-10-11 11:37:03			
End Time: 2001-10-11 11:37:27			
List Schedule	Create Schedule	Show Log	

Note: You can also display the Schedule Runnow Log window by clicking on the process ID (for example, P0t5u6hk1pc). This runs rcaster_logplist.jsp.

- b. Click Delete (runs rcaster_logpdelete.jsp) to delete a selected Process ID.
 - c. Click Delete All (runs rcaster_logpdelete.jsp) to delete all Process IDs.
3. To delete a scheduled job (for example, Turnover Ratio by Branch) from the Log Summary report, select the Schedule ID of that job and click Delete (runs rcaster_logdelete.jsp).

- To delete all scheduled jobs from the Log Summary report, click Delete All (runs rcaster_logdelete.jsp). The Log Summary window displays without any reports:

Log Summary

Please select schedule and click the button.

List Schedule	Create Schedule	Show Log	Log Summary <input type="button" value=""/>
Processes Delete Delete All			
Schedule ID	Job Name	Job Description	
Processes Delete Delete All			
List Schedule	Create Schedule	Show Log	

CHAPTER 6

ReportCaster API Servlet Samples

Topics:

- Sample HTML Calling Forms
- Logging on to the ReportCaster Servlet API
- Servlet Samples

A set of HTML sample forms that use servlet functionality are distributed with the ReportCaster API in the `ibi_html\broker` subdirectory under the WebFOCUS installation directory. Use these sample forms or customize them for your application needs.

Sample HTML Calling Forms

The following sample HTML forms are distributed with the ReportCaster Servlet API:

Form	Servlet	Description	Go To Page
rbasched.htm	DSTSCHED	Schedules a job.	6-6
rbastats.htm	DSTACTIVE	Sets the status of a scheduled job.	6-12
rbarunow.htm	DSTRUNNOW	Runs a scheduled job immediately.	6-13
rbabulk.htm	DSTDLBULK	Maintains a Distribution List with member names from an external file.	6-14
rbabulkm.htm	DSTDLBULK	Maintains a Distribution List with member names entered in an HTML text box.	6-15
rbadlmem.htm	DSTDLMEM	Maintains a Distribution List for a single member.	6-17
rbacopy.htm	DSTDLBULK	Maintains a Distribution List with member names from an existing Distribution List.	6-18
rbadlist.htm	DSTDLLIST	Displays a Distribution List in a browser.	6-19
rbalog.htm	DSTLOG	Displays a log report in the browser.	6-20

Logging on to the ReportCaster Servlet API

The ReportCaster API requires a valid WebFOCUS logon. A sample logon form is distributed with ReportCaster in the following location:

drive: \libi\WebFOCUS*rel*\libi_html\broker\rbalagon.html

where:

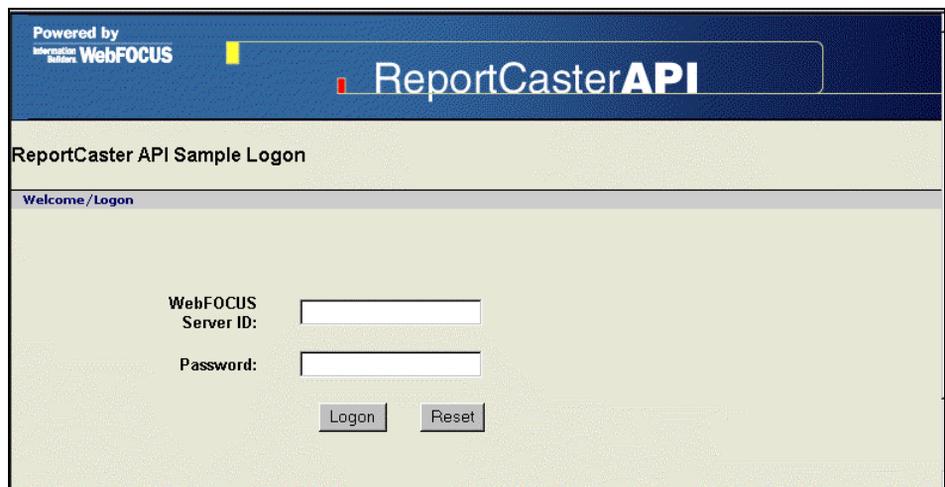
drive

Is the letter of the drive on which WebFOCUS is installed.

rel

Is the WebFOCUS release.

The ReportCaster API Sample Logon screen displays as follows:



Powered by
WebFOCUS

ReportCasterAPI

ReportCaster API Sample Logon

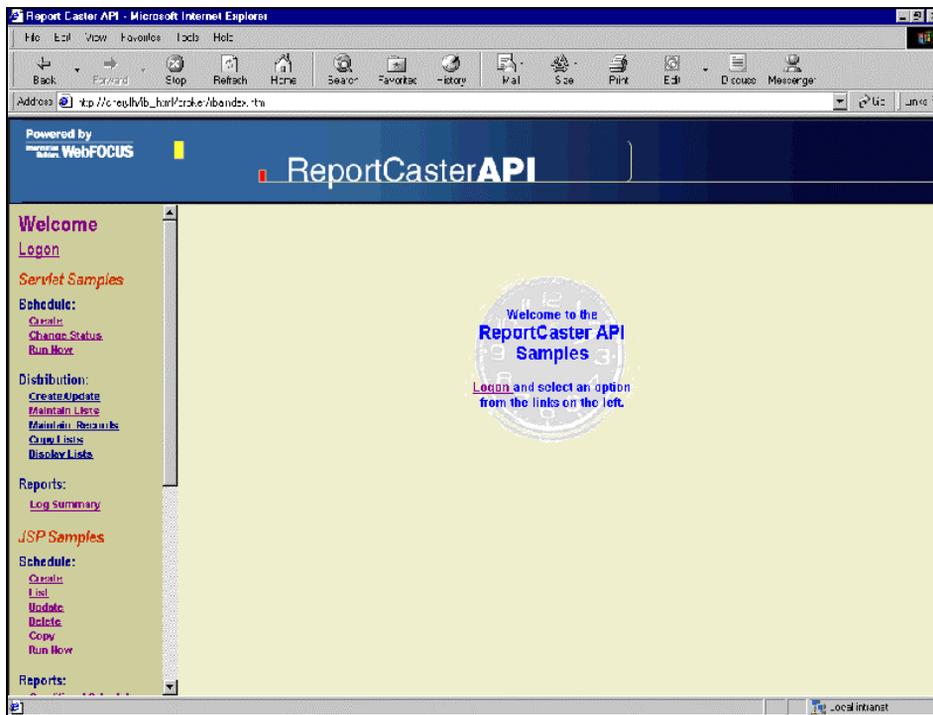
Welcome / Logon

WebFOCUS
Server ID:

Password:

Logon Reset

After you enter a valid WebFOCUS Server ID and password, the ReportCaster API Welcome page displays:



The Welcome page contains links to:

- Servlet Samples. For more information, see *Servlet Samples* on page 6-5.
- JSP Samples. For more information, see Chapter 5, *ReportCaster API JSP Samples*.
- ReportCaster Repository Reports using a WebFOCUS servlet or a WebFOCUS CGI call. For more information, see Chapter 7, *ReportCaster Repository Reports*.

Servlet Samples

Once you have logged into the ReportCaster API, you can click on the following Servlet Samples on the left side of the API window:



Servlet Schedule Options

The Servlet Schedule options enable you to:

- Create a schedule.
- Change the status of an existing schedule.
- Run an existing schedule immediately.

Procedure How to Create a Schedule Using a Servlet

1. Click Create in the Schedule section of the Servlet Samples. This accesses the rbasched.htm form, which runs the DSTSCHED servlet. The Create a Schedule window displays:

Create a Schedule

Required fields are in **bold**. Place mouse over  for help on each field.

Description	
 Job name:	<input type="text" value="centhrv"/>
 Job description:	<input type="text" value="Turnover Ration by Brand"/>
 Pre-processing:	<input type="text"/>
 Post-processing:	<input type="text"/>
 Active:	<input checked="" type="radio"/> Yes <input type="radio"/> No

2. Enter the following values in the Description section:
 - a. **Job name** (required). The name of the report request to be scheduled for execution and distribution. This must be specified in EDAPATH.
Do not include the filename extension. The first character of this value must be alphanumeric. Single quotes and ampersands are not allowed.
 - b. **Job description** (optional). A unique, user-supplied description for the report request (job) being scheduled. Alphanumeric characters with embedded blanks and special characters are allowed.
 - c. **Pre-processing** (optional). The name of a procedure and its parameter string that will run prior to the scheduled report request. This procedure typically performs setup tasks.
 - d. **Post-processing** (optional). The name of a procedure and its parameter string that will run after the scheduled report request.
 - e. **Active** (required). The status of a report request. Valid values are Yes (Active) and No (Inactive).

3. Scroll to the Frequency section and enter the frequency parameters:

Frequency	
Interval:	<input checked="" type="radio"/> Once <input type="radio"/> Hourly <input type="radio"/> Daily <input type="radio"/> Weekly <input type="radio"/> Monthly <input type="radio"/> Yearly
# times within Interval:	<input type="text" value="1"/>
Weekdays: <small>Required if frequency is weekly</small>	<input type="checkbox"/> Mon <input type="checkbox"/> Tue <input type="checkbox"/> Wed <input type="checkbox"/> Thu <input type="checkbox"/> Fri <input type="checkbox"/> Sat <input type="checkbox"/> Sun
Days of the month: <small>Required if frequency is monthly</small>	<input type="text" value="1"/>
Start date (YYYYMMDD):	<input type="text" value="20011010"/>
End date (YYYYMMDD):	<input type="text" value="20991231"/>
Start time (HHMM):	<input type="text" value="1611"/>
End time (HHMM):	<input type="text" value="2359"/>

- a. **Interval** (required). The period of time (the interval) on which report execution and distribution are based. Valid values are:
 - Once
 - Hourly
 - Daily
 - Weekly
 - Monthly
 - Yearly
- b. **# times within interval** (optional). The number of times you want to run your report within the interval specified.
- c. **Weekdays** (optional). If you specified a weekly interval, this parameter is the day of the week the report request will run.

This parameter can occur multiple times. For example, if you specify a weekly interval, and specify 3 for the # of times within interval, the report runs every 3 weeks. Values are required for the specific days of the week such as Monday, Wednesday, and Friday.

- d. **Days of the month** (optional). If you specified a monthly interval, this parameter is the day of the month the report request will run.

This parameter can occur multiple times. For example, if you specify a monthly interval, and specify 2 for the # of times within interval parameter, the report runs every 2 months. Values are required for the specific days in the month, such as the 1st and the 15th.

- e. **Start date (YYYYMMDD)** (optional). The date of first report execution and distribution, in the format *YYYYMMDD*, where *YYYY* is the 4-digit year, *MM* is the month, and *DD* is the day of the month.
- f. **End date (YYYYMMDD)** (optional). The date of last report execution and distribution, in the format *YYYYMMDD*, where *YYYY* is the 4-digit year, *MM* is the month, and *DD* is the day of the month.
- g. **Start time (HHMM)** (optional). The time of first report execution and distribution, in the format *HHMM*, where *HH* is the hour and *MM* is the minute.
- h. **End time (HHMM)** (optional). The time of last report execution and distribution, in the format *HHMM*, where *HH* is the hour and *MM* is the minute.

- 4. Scroll to the Send to section and enter the send to parameters:

Send to	
Distribution list:	<input type="text"/>
Burst report:	<input type="radio"/> Yes <input checked="" type="radio"/> No

- a. **Distribution list** (required). An existing Distribution List.
- b. **Burst Report**. Possible values are Yes (burst) and No (no burst).

5. Scroll to the Send method section and enter the send method parameters:

Send method

Send via: Email FTP Printer

Send format: HTML ▾

E-mail: Mail server: smtpserver
Required if distribution method is via e-mail

Reply Address: chuck_hill@ibi.com

Subject: _____

Company: _____

FTP: FTP server: _____
Required if distribution method is via FTP

Location: _____

User: _____

Password: _____

Index file name: _____

Printer: _____

- a. **Send via** (required). The distribution method. Possible values are e-mail, FTP, or printer.
- b. **Send format** (required). The format of your distributed report output. Possible values are:
 - WP for plain text.
 - HTML for hypertext (recipient must have a browser in order to view).
 - PS for PostScript (recipient must have a GhostView application in order to view).
 - DOC for plain text with ASCII page breaks.
 - PDF for Adobe Acrobat PDF format (recipient must have Adobe in order to view).
 - EXCEL (recipient must have EXCEL in order to view).
 - EXL2K (recipient must have EXCEL 2000 in order to view).
 - WK1, a Lotus 1-2-3 spreadsheet format.

Note:

- If you specified a distribution method of e-mail, proceed to step 6.
- If you specified a distribution method of FTP, proceed to step 7.
- If you specified a distribution method of printer, the output is sent to the printer selected in the Distribution List. No parameters are required. Proceed to step 8.

6. If you specified a distribution method of e-mail, enter the following e-mail parameters:
 - a. **Mail server** (required). The name of the mail server that distributes the report. Single quotes and ampersands are not allowed. The default value is the value specified in the ReportCaster configuration file (bkrsched.cfg).
 - b. **Reply Address** (required). The e-mail address for return mail (the reply address). Single quotes and ampersands are not allowed, and an @ character must be included.
 - c. **Subject** (optional). The content or purpose of the e-mail.
 - d. **Company** (optional). The company to which the sender is associated.
7. If you specified a distribution method of FTP, enter the following FTP parameters:
 - a. **FTP server** (required). The name of the FTP server that distributes the report. Single quotes and ampersands are not allowed. The default value is the value specified in the ReportCaster configuration file (bkrsched.cfg).
 - b. **Location** (required). The destination of the FTP-distributed report. It must be a subdirectory of the FTP server root directory, or an alias defined to the FTP server. Single quotes and ampersands are not allowed.
 - c. **User** (required). The user ID authorized for FTP transfer. Single quotes and ampersands are not allowed. The default value is the value specified in the ReportCaster configuration file (bkrsched.cfg).
 - d. **Password** (required). The password associated with the user ID authorized for FTP transfer. Single quotes and ampersands are not allowed. The default value is the value specified in the ReportCaster configuration file (bkrsched.cfg).
 - e. **Index file name** (optional). The name of the FTP index file for a bursted report. The first character must be alphanumeric, and the default value is HOLD.

8. Scroll to the Notification section and enter the notification parameters:

The screenshot shows a web form titled "Notification". It contains the following elements:

- Notify:** A group of radio buttons. The "Inactive" option is selected. Other options are "Notify of Error" and "Always Notify".
- Notify Subject:** A text input field containing the text "Turnover Ration by Brand".
- Notify Address:** A text input field containing the email address "chuck_hill@ibi.com".
- Notify Device Info:** An empty text input field.
- Notify Reply Address:** A text input field containing the email address "chuck_hill@ibi.com".
- At the bottom of the form are two buttons: "Submit" and "Reset".

- a. **Notify** (required). The notification of job status. Possible values are:
 - Inactive (no notification is sent).
 - Always Notify (notification is always sent).
 - Notify of Error (content of the log is sent on an error condition).
 - b. **Notify Subject** (optional). The content or purpose of the notification message. Single quotes and ampersands are not allowed.
 - c. **Notify Address**. Required if Notify is set to Always Notify or Notify of Error. It is the e-mail address of the person who will receive full notification (complete log report).
 - d. **Notify Device Info** (optional). For a device other than e-mail.
 - e. **Notify Reply Address**. Required if Notify is set to Always Notify or Notify of Error. It is the e-mail address for return mail (the response to the notification message). Single quotes and ampersands are not allowed, and an @ character must be included.
9. Click Submit or Reset.

If you click Reset, the screen is cleared.

If you click Submit, the following message displays:

`The ReportCaster transaction has finished successfully`

Procedure

How to Change the Status of a Scheduled Job

1. Click Change Status in the Schedule section of the Servlet Samples. This accesses the rbastats.htm form, which runs the DSTACTIVE servlet. The Change Status of Scheduled Job window displays:

Change Status of Scheduled Job

Required fields are in **bold**. Place mouse over [?](#) for help on each field.

Functionality

[?](#) Job description:
Required if **Schedule ID** is blank

[?](#) Schedule ID:
Required if **Job description** is blank

[?](#) **Active:** Yes No

2. Enter the following functionality parameters:
 - a. **Job description.** The job description of your previously scheduled job. This parameter is required if Schedule ID is blank.
 - b. **Schedule ID.** The Schedule ID of your previously scheduled job. This parameter is required if job description is blank.
 - c. **Active** (required). The status of a report request. Valid values are Yes (Active) and No (Inactive).
3. Click Submit or Reset.

If you click Reset, the screen is cleared.

If you click Submit, the following message displays:

The ReportCaster transaction has finished successfully

Procedure**How to Run a Scheduled Job Immediately**

1. Click Run Now in the Schedule section of the Servlet Samples. This accesses the rbarunow.htm form, which runs the DSTRUNNOW servlet. The Run a Scheduled Job Immediately window displays:

Run a Scheduled Job Immediately

Required fields are in **bold**. Place mouse over  for help on each field.

Description

 Job description:
Required if **Schedule ID** is blank

 Schedule ID:
Required if Job description is blank

 Priority: 1 2 3 4 5

 User ID:

Target Job Parameters

 Sample Job Parameters:

Continent:

Region:

Country:

2. Enter the following Description parameters:
 - a. **Job description.** The job description of your previously scheduled job. This parameter is required if Schedule ID is blank.
 - b. **Schedule ID.** The Schedule ID of your previously scheduled job. This parameter is required if job description is blank.
 - c. **Priority.** Denotes the priority of a report request. 1 is the highest priority and 5 is the lowest. 3 is the default.
 - d. **User ID** (optional). The WebFOCUS Reporting Server user ID of the owner of the scheduled job.
3. Enter the Sample Job parameters (for example, Continent, Region, and Country). These are sample parameters that you can modify for your own needs.
4. Click Submit or Reset.

If you click Reset, the screen is cleared.

If you click Submit, the following message displays:

The ReportCaster transaction has finished successfully

Servlet Distribution Options

The Servlet Distribution options enable you to:

- Create or update a Distribution List from a file.
- Maintain a Distribution List.
- Maintain a single record in a Distribution List.
- Copy one Distribution List to another.
- Display a Distribution List.

Procedure

How to Create/Update a Distribution List From a File

1. Click Create/Update in the Distribution section of the Servlet Samples. This accesses the rbabulk.htm form, which runs the DSTDLBULK servlet. The Create/Update Distribution Lists from a file window displays:

Create/Update Distribution Lists from a file

Required fields are in **bold**. Place mouse over  for help on each field.

Functionality

 **Distribution List:**

 Access: Public Private

 Send method: E-mail FTP Printer

 Function: Create Append Replace

 User id:

Address Details

 **Address file name:**

2. Enter the following functionality parameters:
 - a. **Distribution List** (required). The name of the Distribution List that will be created or modified.
 - b. **Access**. Possible values are Public (the default) and Private.
 - c. **Send method**. Possible values are E-mail, FTP, or Printer.
 - d. **Function**. Possible values are:
 - Create a new Distribution List.
 - Append new members to an existing Distribution List.
 - Replace all members in an existing Distribution List.
 - e. **User ID** (optional). The WebFOCUS Reporting Server user ID of the owner of the Distribution List.
3. **Address file name** (required). The name of the external source file used to supply member names to the Distribution List.
4. Click Submit or Reset.

If you click **Reset**, the screen is cleared.

If you click **Submit**, the following message displays:

The ReportCaster transaction has finished successfully

Procedure

How to Maintain a Distribution List

1. Click **Maintain Lists** in the **Distribution** section of the **Servlet Samples**. This accesses the `rbabulkm.htm` form, which runs the `DSTDLBULK` servlet. The **Maintain Distribution Lists** window displays:

Maintain Distribution Lists

Required fields are in **bold**. Place mouse over  for help on each field.

Functionality	
Distribution List:	<input type="text" value="eastcoast"/>
User id:	<input type="text" value="ch04838"/>
Access:	<input checked="" type="radio"/> Public <input type="radio"/> Private
Send method:	<input checked="" type="radio"/> E-mail <input type="radio"/> FTP <input type="radio"/> Printer
Function:	<input checked="" type="radio"/> Create <input type="radio"/> Delete <input type="radio"/> Append <input type="radio"/> Replace
Individual addresses:	Example: <code>tony_blair@downingstreet.uk,ENGLAND,\$</code> <input style="width: 100%; height: 40px;" type="text" value="chuck_hill@ibi.com,, \$
joe_posada@ibi.com,, \$
roger_mussina@abcd.dom,, \$"/>

2. Enter the following functionality parameters:
 - a. **Distribution List** (required). The name of the Distribution List that will be created or modified.
 - b. **User ID** (optional). The WebFOCUS Reporting Server user ID of the owner of the Distribution List.
 - c. **Access**. Possible values are Public (the default) and Private.
 - d. **Send method**. Possible values are E-mail, FTP, or Printer.
 - e. **Function**. Possible values are:
 - Create a new Distribution List.
 - Delete the Distribution List.
 - Append new members to the existing Distribution List.
 - Replace all members in the existing Distribution List.
 - e. **Individual addresses** (optional). Are members to add, delete, or append to your Distribution List. Valid formats are as follows:

E-mail send method: The e-mail address followed by the field on which the report is burst. If the report is not burst, specify a comma in place of the burst field. A comma and a dollar sign terminate the record.

FTP send method: The FTP file name followed by the field on which the report is burst. If the report is not burst, specify a comma in place of the burst field. A comma and a dollar sign terminate the record.

Printer send method: The printer name followed by the field on which the report is burst. If the report is not burst, specify a comma in place of the burst field. A comma and a dollar sign terminate the record.
3. Click Submit or Reset.

If you click **Reset**, the screen is cleared.

If you click **Submit**, the following message displays:

The ReportCaster transaction has finished successfully

Procedure**How to Maintain a Single Distribution List Record**

1. Click Maintain Records in the Distribution section of the Servlet Samples. This accesses the rbadlmem.htm form, which runs the DSTDLMEM servlet. The Maintain Single Distribution Records window displays:

Maintain Single Distribution Records

Required fields are in **bold**. Place mouse over [?](#) for help on each field.

Functionality

[?](#) **Distribution List:**

[?](#) **User id:**

[?](#) **Function:** Add Delete

[?](#) **Value:**

[?](#) **Destination:**

2. Enter the following functionality parameters:
 - a. **Distribution List** (required). The name of the Distribution List that will be modified.
 - b. **User ID** (required). The WebFOCUS Reporting Server user ID of the owner of the Distribution List being added to or deleted from.
 - c. **Function**. Possible values are:
 - Add a new member to the Distribution List.
 - Delete an existing member of the Distribution List.
 - d. **Value** (optional). The field value on which the report is burst.
 - e. **Destination** (required). The destination of the report or report section (for example, chuck_hill@ibi.com,,,\$).
3. Click Submit or Reset.

If you click **Reset**, the screen is cleared.

If you click **Submit**, the following message displays:

The ReportCaster transaction has finished successfully

Procedure How to Copy a Distribution List

1. Click Copy Lists in the Distribution section of the Servlet Samples. This accesses the rbacopy.htm form, which runs the DSTDLBULK servlet. The Copy Distribution Lists window displays:

Copy Distribution Lists

Required fields are in **bold**. Place mouse over  for help on each field.

Functionality

 **Target Distribution List:**

 Access: Public Private

 Send via: E-mail FTP Printer

 Function: Create Append Replace

 **Source Distribution List:**

 User id:
You may specify a different userid other than yourself to own the new list.

2. Enter the following functionality parameters:
 - a. **Target Distribution List** (required). The name of the target Distribution List. The target Distribution List is the one whose members are being created, appended to, or replaced.
 - b. **Access**. Possible values are Public (the default) and Private.
 - c. **Send via**. Possible values are E-mail, FTP, or Printer.
 - d. **Function**. Possible values are:
 - Create a new Distribution List using members from the external source file.
 - Append members from the external source file to the target Distribution List.
 - Replace the members of the target Distribution List with the members of the external source file.
 - e. **Source Distribution List** (required). The name of the external source file used to supply member names to the target Distribution List. The file must:
 - Reside on EDAPATH (Windows NT and UNIX), or in a partitioned data set (PDS) allocated to ddname EDARPC (OS/390).
 - Have the extension .fex on Windows NT and UNIX, although you do not specify it in this field. For example, you would enter *filename.fex* simply as *filename*.
 - f. **User ID** (optional). The WebFOCUS Reporting Server user ID of the owner of the Distribution List.

3. Click Submit or Reset.

If you click Reset, the screen is cleared.

If you click Submit, the following message displays:

The ReportCaster transaction has finished successfully

Procedure

How to Display a Distribution List

1. Click Display Lists in the Distribution section of the Servlet Samples. This accesses the rbadlist.htm form, which runs the DSTDLLIST servlet. The Display Distribution Lists window displays:

Display Distribution Lists

Required fields are in **bold**. Place mouse over  for help on each field.

Functionality

 **Distribution List:**

2. Enter the name of the Distribution List (for example, South).
3. Click Submit or Reset. If you click Submit, the properties of the selected Distribution List display:

ReportCaster Distribution List

Name: South
Owner: ch04838
Access: Public
Method: MAIL

VALUE	DESTINATION
	joe_tex@abcd.com
	john_doe@ibi.com

Servlet Report Options

The following procedure describes how to generate a log report using the ReportCaster Servlet API.

Procedure

How to Run a Log Summary Report

1. Click Log Summary in the Reports section of the Servlet Samples. This accesses the rbalog.htm form, which runs the DSTLOG servlet. The Log Summary window displays:

Log Summary

Required fields are in **bold**. Place mouse over [?](#) for help on each field.

Functionality

[?](#) Job description:

[?](#) Schedule id:

[?](#) Last job executed: Yes No

[?](#) Start date (YYYYMMDD):

[?](#) End date (YYYYMMDD):

[?](#) Start time (HHMM):

[?](#) End time (HHMM):

2. Enter the following optional functionality parameters:
 - a. **Job description.** The unique description that you supplied to identify the report request (job) when it was scheduled.
 - b. **Schedule id.** A unique, API-generated, 12-digit key assigned to the job when it was scheduled.
 - c. **Last job executed.** Possible values are Yes and No. If you select Yes, only the last executed job will display in the log report.
 - d. **Start date.** The date of first report execution and distribution, in the format *YYYYMMDD*, where *YYYY* is the 4-digit year, *MM* is the month, and *DD* is the day of the month. The report will display schedules with a 'next run time' on or after the date specified.
 - e. **End date.** The date of last report execution and distribution, in the format *YYYYMMDD*, where *YYYY* is the 4-digit year, *MM* is the month, and *DD* is the day of the month. The report will display schedules with a 'next run time' on or before the date specified.
 - f. **Start time.** The time of first report execution and distribution, in the format *HHMM*, where *HH* is the hour and *MM* is the minute. The report will display schedules with a 'next run time' on or after the time specified.
 - g. **End time.** The time of last report execution and distribution, in the format *HHMM*, where *HH* is the hour and *MM* is the minute. The report will display schedules with a 'next run time' on or before the time specified.

3. Click Submit or Reset. If you click Submit, the Job Process Log Report displays:

Job Process Log Report	
Job Description: Sales by State	
Server User: radmin	(BTP1010) Starting worker thread
MRE User: admin	(BTP1010) Method: Mail. Host: 127.0.0.1.
Process: Pot74ok9f01	(BTX000) (FOC43011) FILE hold.htm SUCCESSFULLY DISTRIBUTED
Procedure: app/salesbys	(BTP3080) Resolving Broker Server temporary space for BKRLOG service
Schedule ID: St74obptk1	(BTP3081) Temporary directory for BKRLOG process follows on next line
Start Time: 2001-10-26 11:04:17	(BTP3081) C:\ibi\srv436\wfs\edatemp\ts000216\
End Time: 2001-10-26 11:05:58	(BTY1010) Broker Request app/salesbys Complete.
Job Description: East Coast Sales Report	
Server User: radmin	(BTP1010) Starting worker thread
MRE User: admin	(BTP1010) Method: Mail. Host: 127.0.0.1.
Process: Pot74onvpc2	(BTX000) (FOC43011) FILE hold.htm SUCCESSFULLY DISTRIBUTED
Procedure: app/salesbyc	(BTP3080) Resolving Broker Server temporary space for BKRLOG service
Schedule ID: St74of0sm2	(BTP3081) Temporary directory for BKRLOG process follows on next line
Start Time: 2001-10-26 11:06:16	(BTP3081) C:\ibi\srv436\wfs\edatemp\ts000216\
End Time: 2001-10-26 11:08:05	(BTY1010) Broker Request app/salesbyc Complete.

Note: If you do not specify any parameters, all jobs will display in the Job Process Log Report.

CHAPTER 7

ReportCaster Repository Reports

Topics:

- Running ReportCaster Repository Reports
- Distribution Type Report
- Address Book Report
- Schedule Contacts Report
- No Contacts Report
- ReportCaster Users Report
- Alerts Schedules Report

The following reports access the ReportCaster Repository and are available from the ReportCaster API sample page:

- Distribution Type (rcdlist)
- Address Book (rcabook)
- Schedule Contacts (rcnotify)
- No Contacts (rcnonote)
- ReportCaster Users (rcuser)
- Alert Schedules (rcalert)

There is no security built into these reports. If you run a report to see all Address Books (rcabook), you will see all public and private Address Books. However, you can modify the code of these sample reports to apply the level of security appropriate for your organization.

Running ReportCaster Repository Reports

ReportCaster Repository reports are located in the `ibi/apps/rcutil` subdirectory of the Web server on which WebFOCUS is installed. You can run these reports by:

- Importing the reports into any domain and then adding the reports to the domain's Standard Reports folder.
- Running them from the ReportCaster API sample page located in `ibi/WebFOCUSrel/ibi_html/broker/rbalogon.htm` (where *rel* is the WebFOCUS release). You can run these WebFOCUS Reports using the:
 - WebFOCUS Servlet
 - WebFOCUS CGI



Relational Databases

ReportCaster Repository reports run for all FOCUS repositories without any modifications to the programs. However, when running these sample reports (rcdlist, rcabook, rcnotify, rcnonote, rcuser, rcalert) using a relational database, you must comment out the default code that is used by FOCUS. For example,

```
- *  
- * USE  
- * BOTADDR ON FOCSU01  
- * BOTDEST ON FOCSU01  
- * END
```

UNIX and NT: These procedures are located in the `ibi/apps/rcutil` directory.

OS/390: These procedures are located in `qualif.EDARPC.DATA` (where *qualif* is your high-level qualifier).

In addition, if you are using OS/390, you must also remove the following tables from EDAMFD:

- BOTSCHED
- BOTPARMS
- BOTADDR
- BOTDEST
- BOTLOG
- BOTLOG2
- BOTTELL
- BOTJOURN

Replace these tables by running the Synonym Wizard. For more information about the Synonym Wizard, see your iWay Server documentation.

ReportCaster Repository Reports (OS/390 Specific)

If you are using WebFOCUS Version 4 Release 3.6 Java Gen 251 and earlier, you must remove the .fex from the Style Sheet code before running the rcabook, rcdlist, and rcuser reports on OS/390. For example:

Old rcabook:

```
line 69 TYPE=DATA,FOCEXEC=rcdldet.fex(BOOKNAME=N1),
```

New rcabook:

```
line 69 TYPE=DATA,FOCEXEC=rcdldet(BOOKNAME=N1),
```

Old rcdlist:

```
line 79 TYPE=DATA,FOCEXEC=rcdldet.fex(BOOKNAME=N3),
```

New rcdlist:

```
line 79 TYPE=DATA,FOCEXEC=rcdldet(BOOKNAME=N3),
```

Old rcuser:

```
line 68 TYPE=DATA,FOCEXEC=rcudets.fex(USERID=N1),
```

```
line 72 TYPE=DATA,FOCEXEC=rcabook.fex(USERID=N1),
```

New rcuser:

```
line 68 TYPE=DATA,FOCEXEC=rcudets(USERID=N1),
```

```
line 72 TYPE=DATA,FOCEXEC=rcabook(USERID=N1),
```

These procedures are located in *qualif*.EDARPC.DATA (where *qualif* is your high-level qualifier).

Distribution Type Report

The Distribution Type (rcdlist) report displays the following properties of a Distribution List:

- **List Owner**—The WebFOCUS Reporting Server user ID that created the list.
- **Distribution Type**—FTP, MAIL, or PRNT.
- **Distribution List Name.**
- **Access**—Public or Private.

Example

Running a Distribution Type (rcdlist) Report

When you run the Distribution Type (rcdlist) report from the ReportCaster API sample page, select one of the following parameters:



The screenshot shows a web form titled "Distribution Lists by Type". Below the title is a horizontal line. Underneath, the text "Select type of distribution:" is followed by four radio button options: "All", "Mail", "FTP", and "Printer". The "All" option is selected. At the bottom of the form are two buttons: "Run Report" and "Clear".

In the following display, ALL is entered as the distribution type:

PAGE 1			
ReportCaster Address Book Distribution Lists			
<i>List Owner</i>	<i>Distribution Type</i>	<i>Distribution List Name</i>	<i>Access</i>
Caster1	MAIL	mail list	Public
	PRNT	printers	Public
mb03324	FTP	FTPLIST	Public
	MAIL	email list1	Public
		mblast1	Public
		mblast2	Private
	PRNT	print list1	Public

Once the report displays, you can drill-down on the Distribution List name, which then displays the addresses and burst values (if any) of that Distribution List. The following displays a drill-down on the mail_list Distribution List name:

PAGE 1	
ReportCaster Address Book Addresses Assigned to Distribution List mail_list	
<i>Addresses</i>	<i>Burst Value</i>
john_smith@ibi.com	
mark_brown@ibi.com	

Address Book Report

The Address Book (rcabook) report displays the Distribution Lists and methods owned by a particular WebFOCUS Reporting Server user ID. When you run the report, you will be prompted for the user ID (for example, mb03324). The report then displays as follows:

PAGE 1		
ReportCaster Address Books Owned by mb03324		
<i>Distribution List Name</i>	<i>Distribution Type</i>	<i>Access</i>
FTPLIST	FTP	Public
email list1	MAIL	Public
mblast1	MAIL	Public
mblast2	MAIL	Private
print list1	PRNT	Public

The Address Book report also enables you to drill-down on the Distribution List name, which then displays the addresses and burst values (if any) of that Distribution List.

Schedule Contacts Report

The Schedule Contacts (rnotify) report is sorted by user ID (Owner ID). It also displays the schedule ID, job description, type of notification (full or brief), notify reply address, and the notification status of a schedule.

PAGE 1						
Notification Information						
<i>Owner id</i>	<i>Schedule id</i>	<i>Job Description</i>	<i>Full Notification</i>	<i>Brief Notification</i>	<i>Notify Reply Address</i>	<i>Notification Status</i>
mb03324	St2enrce1	methodof	mark_brown@bi.com		mark_brown@bi.com	Always

No Contacts Report

The No Contacts (rcnonote) report is sorted by user ID (Schedule Owner). It also displays schedule ID, job description, and the notification status of a schedule.

PAGE 1			
Schedules That Do Not Send Notification			
<i>Schedule Owner</i>	<i>Schedule id</i>	<i>Job Description</i>	<i>Notification Status</i>
mb03324	St2hvatau1	rcabook	Never
	St2i0qu0e2	Sales Report	Never

ReportCaster Users Report

The ReportCaster Users (rcuser) report displays a list of all valid user IDs on the WebFOCUS Reporting Server.

PAGE 1		
ReportCaster Users		
<i>WebFOCUS Reporting Server User IDs</i>	<i>Schedules</i>	<i>Address Books</i>
Caster1	Schedules	Address Books
mb03324	Schedules	Address Books

In addition, you can drill-down on the schedules and Address Books of that user ID.

- When you drill-down by clicking Schedules, a list of schedule IDs and job descriptions display.
- When you drill-down by clicking Address Books, the Distribution List names, distribution types (FTP, MAIL, PRNT), and access (public or private) display. You can then drill-down on the Distribution List name to display the addresses and burst values (if any) of that Distribution List.

Alerts Schedules Report

The Alerts schedules (rcalert) report displays the owner (WebFOCUS Reporting Server user ID), schedule ID, job description, and Distribution List name of all ReportCaster Alerts that are scheduled.

PAGE 1			
ReportCaster Alert Schedules			
<i>Owner</i>	<i>Schedule ID</i>	<i>Job Description</i>	<i>Distribution List</i>
mb03324	St4ktlc311	Car Sales Rpt Exceeding 18000	mblist1

CHAPTER 8

Self-Service Two-Way Email Overview

Topics:

- Installation Requirements
- Using the Self-Service Two-Way Email Beans
- Security for Self-Service Two-Way Email
- Processing Overview Using JSP Technology
- Self-Service Two-Way Email Interface
- Self-Service Two-Way Email Administration

Self-Service Two-Way Email works outside of MRE. It allows a developer to use JavaServer Pages technology (or other Java technology) with JavaBeans components to create an independent, customized environment that offers the features of Two-Way Email.

Installation Requirements

Self-Service Two-Way Email requires the following on Windows NT/2000, UNIX, and OS/390 UNIX System Services platforms:

- ReportCaster Version 4 Release 3.6. The Distribution Server is required for Self-Service Two-Way Email.
- Java Development Kit (JDK) 1.3 or later.
- Servlet-enabled Web server or application server.
- SMTP/POP3-enabled Mail server.
- ServletExec™ 4.1.

Important:

Install ServletExec to the default location. Otherwise, you will have problems uninstalling it later.

Procedure

How to Allocate Mail Classes for Self-Service Two-Way Email

Important:

You must perform the following manual steps:

1. Create a new folder under the javaassist directory. You can name this folder anything you want.
2. Copy the ibi/WebFOCUS436/ibi_html/javaassist/meta-inf folder and the ibi/WebFOCUS436/ibi_html/javaassist/com/sun/mail folder into this new directory.

Do not delete the original folders from their locations. They are referenced by other features.
3. In your servlet engine, add the location of this new folder (ibi/WebFOCUS436/ibi_html/javaassist/*new folder*) as a classpath.

Using the Self-Service Two-Way Email Beans

The Two-Way Email API consists of JavaBeans components. These JavaBeans components (or Beans) handle the application logic for a Self-Service Two-Way Email application.

You cannot directly access the Beans from a directory. They live in the DSTServlets.jar file, located by default in the WebFOCUS javaassist directory, and are unpackaged at run-time.

You supply property values for the Beans from the input collected on the interface, as described in Chapter 9, *Self-Service Two-Way Email Samples*.

Self-Service Two-Way Email Beans Within the `ibi.broker.beans` and `ibi.broker.beans.handler` Packages

The following table describes the Self-Service Two-Way Email Beans within the `ibi.broker.beans` package.

Bean Name	Description
DSTAuthenticate	Contains the property information needed for security authentication to the Distribution Server.
DSTBeanHandler	Is the super class of all the Beans that process requests.
DSTBeanResult	Contains results of all requests processed by the Beans.
DSTAPIStatus	Contains error messages or codes returned from all requests processed by the Beans.
DSTTowwayFactory	Contains all the property information needed to create a Self-Service Two-Way Email template.

The following table describes the Self-Service Two-Way Email Bean within the `ibi.broker.beans.handler` package.

Bean Name	Description
DSTTWSelfHandler	Processes all Self-Service Two-Way Email requests.

For detailed online documentation about these Beans, developers can access:

- http://hostname/ibi_html/broker/docapibeans/index.html (where *hostname* is the host name of the Web server on which you installed WebFOCUS).
- *ReportCaster API for JavaBeans Components* on your documentation CD.

Procedure

How to Use the Self-Service Two-Way Email Beans

A Self-Service Two-Way Email application requires the following development steps. This procedure applies to development using JSP technology and JavaBeans components on Windows NT/2000. Modify the procedure as needed for your development tool or language.

1. Import Bean packages to make them available to your Java program (JSP, Java servlet, or Java application).

```
<%@ page language="java" import="ibi.broker.beans.*" %>
```

2. Create a DSTAuthenticate object that contains all the property information needed for security authentication to the Distribution Server.

```
<jsp:useBean id="auth"
scope="session" class="ibi.broker.beans.DSTAuthenticate" />
```

3. Create an object for the Bean from which you want to perform functions (for example, creating a Two-Way Email template). In the following code, the factory object contains all the property information needed for Two-Way Email.

```
<jsp:useBean id="factory" scope="session"
class="ibi.broker.beans.TwoWayFactory" />
```

4. Submit an authentication object into the object created in step 3.

If you are using the Common Graphical Interface (CGI):

```
<%
    factory.setAuthenticate(auth);
    factory.clear();
    auth.setRequest(request);
%>
```

If you are not using the CGI:

```
factory.setAuthenticate(auth);
```

5. Specify connection and configuration information.

If you are using the Common Graphical Interface (CGI):

Log on to the Reporting Server using IBIC_user and IBIC_pass in a logon page such as rbalogon.htm or rbaindex.htm (see Chapter 5, *ReportCaster API JSP Samples*). Successful logon creates a WebFOCUS cookie (WF_COOKIE).

```
<jsp:setProperty name = "auth" property="initParameter"
value="drive:/ibi/client51/conf/web/cgi" />
```

where:

drive

Is the letter of the drive on which the WebFOCUS Client is installed.

If you are not using the CGI:

Create a logon page that collects the node, port, user, and password.

For example, using the Java language:

```
<%
String Node = request.getParameter("IBIB_node");
    // Agent_node for distribution
String Port = request.getParameter("IBIB_port");
    // Agent_port for distribution
String User = request.getParameter("IBIB_user");
    // WebFOCUS Reporting Server ID
String Pass = request.getParameter("IBIB_pass");
    // WebFOCUS Reporting Server password
auth.setAgentProperties(Node+":"+Port);
auth.setUser(User);
auth.setPass(Pass);
%>
```

6. For each function (or method) you want to run, perform the following:

- a. Using the object created in step 3, set the properties for the function you want to perform. The following is an example of using the values on an HTML form to set the properties for the factory object.

```
<jsp:setProperty name = "factory" property="*" />
```

- b. Create a request object in the DSTBeanHandler that will contain the function specified in the factory object.

```
DSTBeanHandler app = myBean.getTWSelfHandler
```

- c. In the DSTBeanHandler object, issue the processRequest method to run the Bean function specified in the factory object. For example, to create a new Two-Way Email template, process the request object in Step b as follows:

```
app.processRequest ();
```

- d. Check the status. Issue the `getAPIStatus` method of the `DSTBeanHandler` object to obtain the `DSTAPIStatus` Bean. This describes the status of your request.

```
DSTAPIStatus status = app.getAPIStatus();
int code = status.getErrorCode();
```

or

```
String message = status.getErrorMessage();
```

- e. For functions that return data, such as log functions, you must first retrieve the error code of the `DSTAPIStatus` Bean to obtain the return code. Next, compare the return code to the variable `DSTAPIStatus.NO_ERROR`. There are two possible outcomes:
- The return code and the `DSTAPIStatus.NO_ERROR` variable are equal, meaning that your request was successful and you can now obtain your answer set.
 - The return code and `DSTAPIStatus` are not equal, meaning that your request failed. Use the `getErrorMessage` method in the `DSTAPIStatus` object to obtain a detailed message.

The following section of code illustrates the error recovery process.

```
if(code == DSTAPIStatus.NO_ERROR) {
    result = app.getBeanResult();
    message = "Successfully sent out the Two Way Email message!";
}
else {
    message = status.getErrorMessage();
}
if (message == null)
{
    message = "Something wrong...";
}
```

Note: Messages implemented for the `DSTAPIStatus` object are not currently internationalized.

- f. To obtain an answer set, issue the `getBeanResult` of the `DSTBeanHandler` object. This will return a `DSTBeanResult` object. The result set can be obtained in XML or sent to the browser in HTML format.
- To obtain your result as XML, use the `getXML` method of the `DSTBeanResult`.
 - To send your result to the browser, code a JSP. HTML code incorporated in the JSP is used to display the result set. For more information about the methods and properties that may be accessed in your JSP, see http://hostname/ibi_html/broker/docapibeans/index.html (where *hostname* is the host name of the Web server on which you installed WebFOCUS) and *ReportCaster API for JavaBeans Components* on your documentation CD.

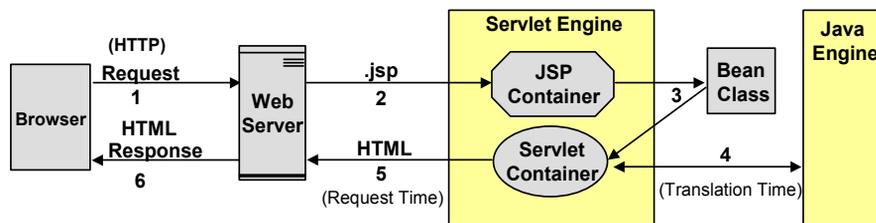
Security for Self-Service Two-Way Email

Self-Service Two-Way Email requires a valid WebFOCUS Reporting Server user ID and password to validate a request and send it to the Distribution Server. The user ID and password are collected on a logon page as IBIC_user and IBIC_pass.

Successful logon to the WebFOCUS Reporting Server creates a WebFOCUS cookie (WF_COOKIE) that contains the validated user ID of the current user. However, the user ID and password can also be set using the Bean properties.

Processing Overview Using JSP Technology

The following diagram illustrates the processing that takes place using JSP technology:



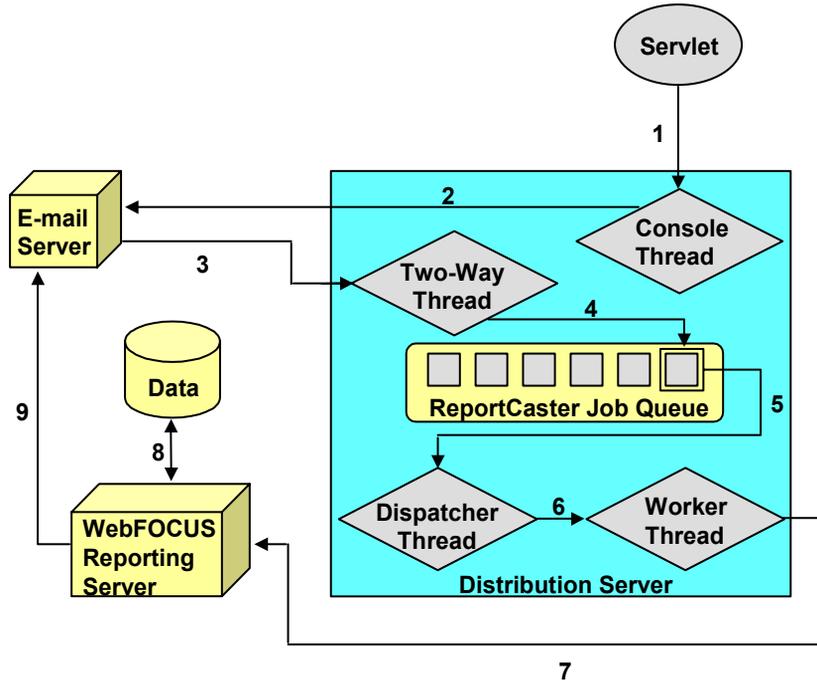
1. A JSP is invoked through a URL in the browser. The page is sent to the Web server using HTTP.
2. The Web server detects the file type of .jsp and sends the JSP to the Servlet engine (for example, JDK 1.3.1).

Note: New Atlanta ServletExec is used as a plug-in to the IIS Web server in order to communicate to the Servlet engine. ServletExec includes a JSP container and a servlet container.

3. The JSP accesses the Bean class and passes the information to the servlet container.
4. The servlet is then compiled by the Java engine and loaded into the servlet container. This process is known as the translation time.
5. The servlet runs and generates HTML, which is sent back to the Web server. This process is known as the request time.
6. The Web server sends the HTML back to the browser.

Processing Overview for a Self-Service Two-Way Email Request

The following diagram illustrates the processing that takes place for a Self-Service Two-Way Email request:



1. When a Self-Service Two-Way Email request is processed, an XML document is created and sent from the servlet to the Distribution Server.
2. The Distribution Server's console thread, always listening for a request from the servlets, converts the XML document into an e-mail and sends it to the E-mail Server specified in the request.
3. Once a user responds to a Self-Service Two-Way Email template, the Two-Way Email listener on the Distribution Server gets the request.
4. The Two-Way Email listener places the job on the ReportCaster queue.
5. A dispatcher thread picks up the job from the queue.
6. A worker thread picks up the job from the dispatcher thread.
7. The worker thread begins communication with the WebFOCUS Reporting Server.
8. The WebFOCUS Reporting Server accesses the data it needs to run the job.
9. The WebFOCUS Reporting Server converts the report into a simple e-mail that is sent to the user's inbox.

Self-Service Two-Way Email Interface

A Self-Service Two-Way Email application requires an interface to gather the information required by Two-Way Email and create a session with the Distribution Server. You can create the Self-Service Two-Way Email interface using JSP technology, a Java servlet, or a Java application.

Information Builders provides a sample interface that uses JSP technology. For more information, see Chapter 9, *Self-Service Two-Way Email Samples*.

Self-Service Two-Way Email Administration

A Self-Service Two-Way Email application requires an Administrator's console to enable administrators to monitor the execution of report requests, cancel requests, and perform other administrative tasks.

For more information, see Chapter 10, *Self-Service Two-Way Email Administrator Console*.

CHAPTER 9

Self-Service Two-Way Email Samples

Topics:

- Self-Service Two-Way Email JavaServer Pages
- Logging on to Self-Service Two-Way Email
- Creating a Self-Service Two-Way Email Template

Sample Self-Service Two-Way Email JavaServer Pages (JSP) that use Bean functionality are distributed with WebFOCUS in the `ibi_html\mobile` subdirectory under the WebFOCUS installation directory. Use these samples or customize them for your application needs.

These sample pages require a valid WebFOCUS logon and use *localhost* and port *8200* to obtain configuration information. If you are using a different host and port, you must manually change these values in the sample pages that perform authentication.

Self-Service Two-Way Email JavaServer Pages

The following table lists the Self-Service Two-Way Email sample JavaServer Pages.

Self-Service Two-Way Email JavaServer Pages	Description	Bean Properties/Method	Go To Page
tw_twoway.jsp	Input form to create a template.	Required properties for DSTTwoWayFactory Bean: <ul style="list-style-type: none">• IBIB_fex• IBIB_format• IBIB_toaddr• IBIB_reply• IBIB_subject• IBIB_message	9-5
tw_focexec.jsp	Input form for FOCEXEC.	Required properties for DSTTwoWayFactory Bean: None	9-7
tw_summary.jsp	Updates template information.	Method: getTWSelfHandler	9-7

Logging on to Self-Service Two-Way Email

Self-Service Two-Way Email requires a valid logon to the WebFOCUS Reporting Server.

- If you installed only Self-Service Two-Way Email, you can use the sample logon form `twlogon.htm`, which on Windows NT/2000 resides by default in:

`drive:\ibi\WebFOCUSrel\ibi_html\mobile`

where:

`drive`

Is the letter of the drive on which WebFOCUS is installed.

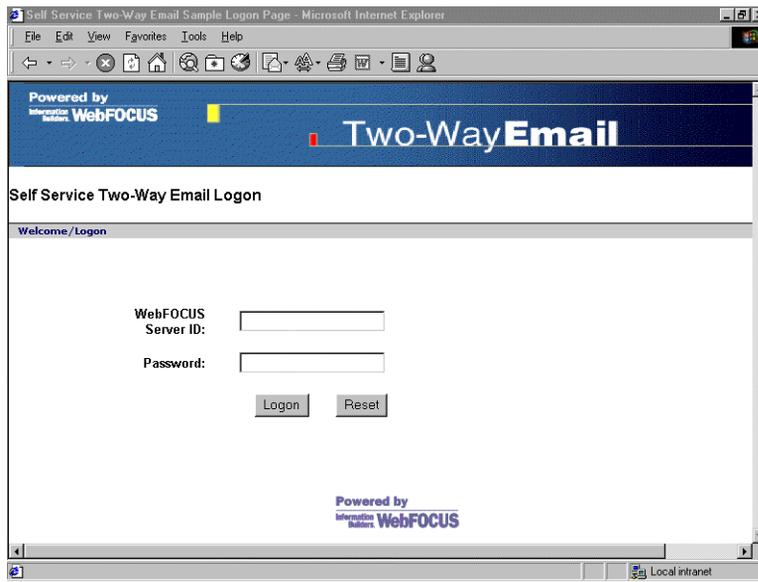
`rel`

Is the WebFOCUS release.

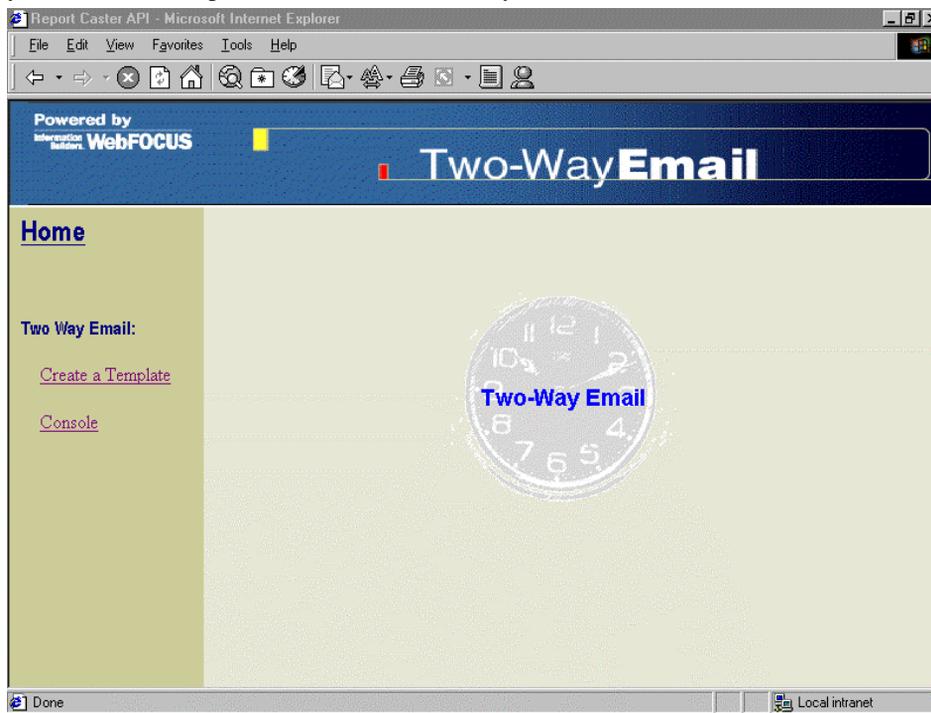
- If you installed Two-Way Email with any other license code, use `mrlogon.htm`, which on Windows NT/2000 resides by default in:

`drive:\ibi\WebFOCUSrel\ibi_html\workbnch`

The `twlogon.htm` page displays as follows:



Once you successfully log on, you are directed to `twindex.htm`. This sample page enables you to create a template or access the Two-Way Email Administrator console.



Creating a Self-Service Two-Way Email Template

The Create a Template sample page (`tw_twoway.jsp`) provides an interface that gathers the information needed to create and send a template to a Self-Service Two-Way Email subscriber. Once the template is sent successfully, an e-mail message containing that template arrives at the inbox of the subscriber's account. When the subscriber replies to the message, supplying parameter values if required, Self-Service Two-Way Email sends another message containing the report.

As a result of template creation, Self-Service Two-Way Email creates an XML document that is sent to the Distribution Server. For more information about the processing that takes place for a Self-Service Two-Way Email request, see Chapter 8, *Self-Service Two-Way Email Overview*.

Procedure**How to Create a Self Service Two-Way Email Template**

1. Click Create a Template (runs `tw_twoway.jsp`). The Self Service Two-Way Email template creating Tool window displays:

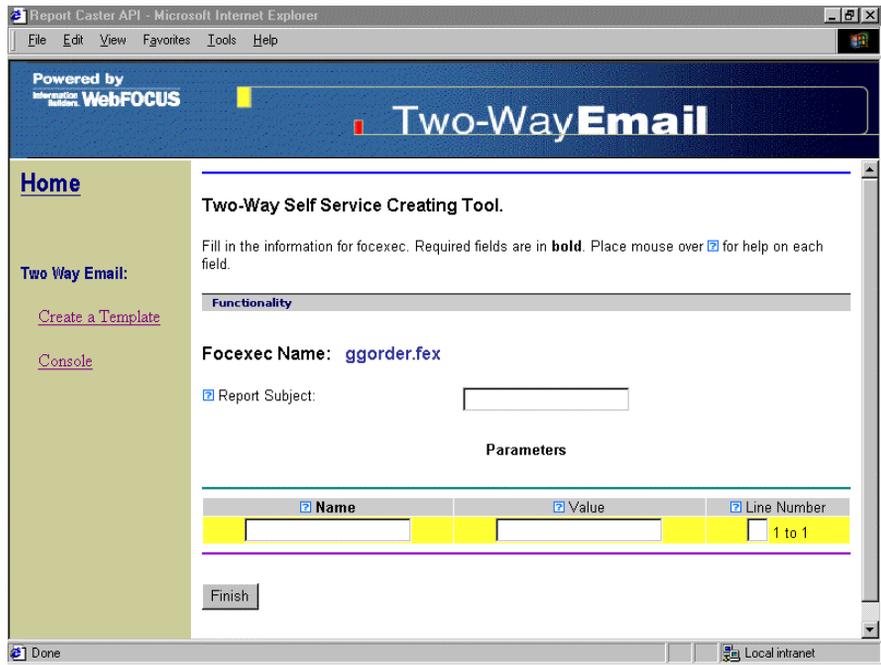
The screenshot shows a web browser window titled "Report Caster API - Microsoft Internet Explorer". The page header includes "Powered by WebFOCUS" and "Two-Way Email". The main content area is titled "Self Service Two-Way Email template creating Tool." and contains a form with the following fields:

- Functionality**
 - Please enter focexec file:
 - Do you need parameters for this focexec? Yes No
 - Number of parameters:
 - Format of the report: text html
 - Subscriber's email:
 - Reply email:
 - Subject:
 - Message:

The left sidebar contains a "Home" link and a "Two Way Email:" section with links for "Create a Template" and "Console".

2. Enter the following functionality parameters:
 - a. **Please enter focexec file** (required). This procedure must reside on EDAPATH. The name must include the extension `.fex` (for example, `nesales.fex`). This value populates the `IBIB_fex` property.
 - b. **Do you need parameters for this focexec?**
Yes, if the procedure requires one or more parameter values.
No, if the procedure does not require any parameter values.
 - c. **Number of parameters** (optional). A numeric value representing the number of parameters in the procedure.
 - d. **Format of the report** (required). The report display format (for example, text or HTML). This value populates the `IBIB_format` property.
 - e. **Subscriber's email** (required). The e-mail address of the Self-Service Two-Way Email subscriber. The template will be sent to this address. This value populates the `IBIB_toaddr` property.
 - f. **Reply email** (required). The address that the Two-Way Email Server is listening on. This address is in the `ACCOUNT` parameter in the `TWO-WAY` section of the `bkrsched.cfg` file. For more information on the `bkrsched.cfg` file, see the *WebFOCUS ReportCaster Administrator's Manual*.
This value populates the `IBIB_reply` parameter.

- g. **Subject** (required). The subject shown in the e-mail containing the Two-Way Email template. This value populates the IBIB_subject property.
 - h. **Message** (required). The message shown in the e-mail containing the Two-Way Email template. This message tells the recipient the information required to run the associated report. For example: Please enter the product on the first line of your reply and the region on the second line.
This value populates the IBIB_message property.
 - i. **Year (YYYY), Month (MM), Day (DD)** (optional). These three parameters specify the expiration date of the current template, in the format YYYY-MM-DD. After this date, the user can no longer request the template.
3. Click Next (runs tw_focexec.jsp) to access the Two-Way Self Service Creating Tool window.



4. Enter the following functionality parameters for the procedure specified in step 2:
 - a. **Report Subject** (optional). The subject of the e-mail containing the report. This value populates the IBIB_repsub property.
 - b. **Name** (optional). The name of a parameter required by the procedure (for example, PRODUCT or REGION). This value populates the IBIB_parname property.
 - c. **Value** (optional). A value for the parameter. If you supply a value, do not include a Line Number. This value populates the IBIB_parvalue property.
 - d. **Line Number** (optional). The input line number of a reply, on which the user dynamically enters a value for a parameter.
A Line Number cannot exceed the number of parameters used in the procedure.
A Line Number is required if you do not supply a value for a parameter.
This value populates the IBIB_position parameter.
5. Click Finish (runs tw_summary.jsp).

CHAPTER 10

Self-Service Two-Way Email Administrator Console

Topics:

- Accessing the Administrator Console
- Using the Job Log
- Using the Event Log
- Checking the Status of a Job or Canceling a Job

A Self-Service Two-Way Email application requires an Administrator Console to enable administrators to monitor the execution of report requests, cancel requests, and perform other administrative tasks.

If you installed only Self-Service Two-Way Email:

- The Two-Way Email Administrator Console consists of the Job Log, Event Log, and Job Status tabs.
- The `TWOWAY` parameter in the TWO-WAY section of the `bkrsched.cfg` file is set to `SELF`.
- You must log on using your WebFOCUS Reporting Server credentials, which create a WebFOCUS cookie.

If you installed Two-Way Email with any other license code:

- The Two-Way Email Administrator Console consists of the Users, Job Log, Event Log, and Job Status tabs.
- The `TWOWAY` parameter in the TWO-WAY section of the `bkrsched.cfg` file is set to `YES`.
- You must log on using your WebFOCUS Reporting Server credentials, which create a WebFOCUS cookie. Additionally, you must create an MRE cookie when you log on using the logon page `mrlogon.htm`. For information about the MRE console, see the *WebFOCUS Managed Reporting Administrator's Manual*.

Accessing the Administrator Console

Once you have successfully logged onto Self-Service Two-Way Email (see Chapter 9, *Self-Service Two-Way Email Samples*, for details), you are directed to `twinde.htm`. Access the Two-Way Email Administrator Console by clicking the Console link on this page.

Using the Job Log

The Job Log displays information about the activities that occurred during the execution and distribution of a report. It confirms that a report was executed and distributed successfully. If a job is not successful, the Job Log states the reason why.

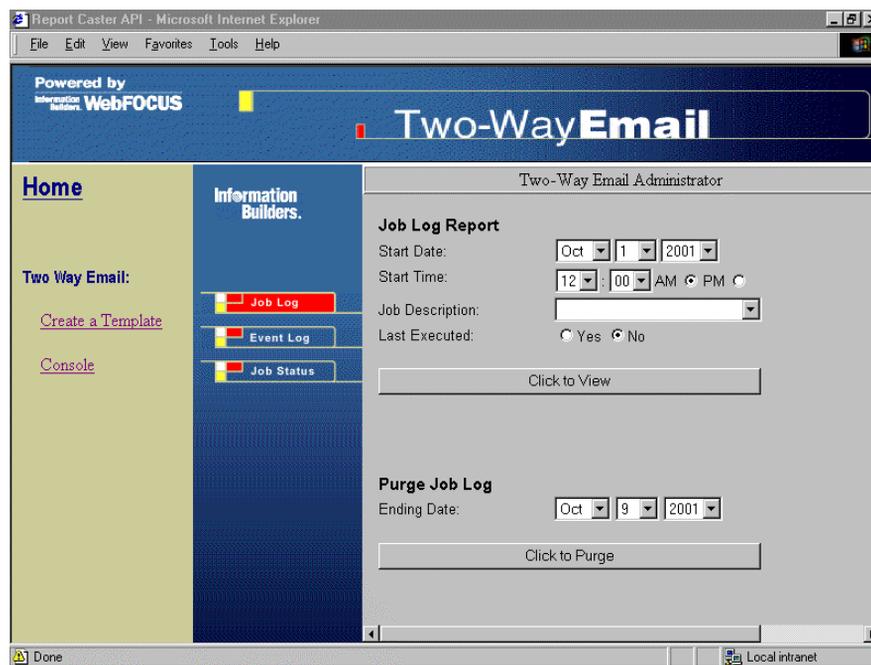
A Job Log includes the following:

- **Job Description.** The name of the report (procedure) associated with the job.
- **Server User.** The WebFOCUS Reporting Server user ID.
- **Process.** A unique, system-generated key that identifies a specific execution of the report.
- **Job Name.** The name of the report.
- **Start Time.** The date and time the job started running.
- **End Time.** The date and time the job finished running.
- **Messages.** Details on the activities that took place during job execution and distribution.

Procedure**How to View the Status of an Executed Job**

1. From the Two-Way Email Administrator Console, click the Job Log tab.

Note: The following screen displays if you have installed only Self-Service Two-Way Email.



2. Apply selection criteria to define the information that will be retrieved:
 - a. Use the drop-down lists for Start Date and Start Time to specify the beginning of the timeframe that you are interested in. Information on all jobs run from that date and time up to the current date and time will be retrieved.
 - b. Type the name of the report in the Job Description field. Information only for the jobs associated with that report during the specified timeframe will be retrieved.
 - c. Select Yes for Last Executed. Information only for the most recently executed job (Job Description) during the specified timeframe will be retrieved.
3. Click Click to View to display the Job Log report.

Purging the Job Log

Since the Job Log accumulates information and can become difficult to navigate, it is a good idea to periodically purge it to conserve space.

You can control automatic purging of the Job Log at predefined intervals of time. Edit the LOG_PURGE_PERIOD parameter in the SCHEDULE section of the Distribution Server's bkrsched.cfg file.

Example

Purging Log Reports Every Two Days

To automatically purge Job Log reports every two days, enter the following in the SCHEDULE section of the Distribution Server's bkrsched.cfg file:

```
LOG_PURGE_PERIOD 2
```

This parameter applies to the Self-Service Two-Way Email Job Log and the ReportCaster Job Log. For more information about the bkrsched.cfg file, see the *WebFOCUS ReportCaster Administrator's Manual*.

Procedure

How to Purge the Job Log

1. From the Two-Way Email Administrator Console, click the Job Log tab.
2. Using the drop-down lists for Ending Date, specify a date through which the Job Log will be purged.
3. Click **Click to Purge**. Respond to the confirmation message on the next dialog box. Self-Service Two-Way Email will delete all reports from the beginning of the Job Log through the specified ending date.

Using the Event Log

Use the Event Log to monitor the status of a Self-Service Two-Way Email request as it is received, checked for security, sent to the WebFOCUS Reporting Server for execution, and distributed as a report.

You can customize the content of the Event Log. Edit the RECOVERY parameter in the SCHEDULE section of the Distribution Server's bkrsched.cfg file as follows.

Setting	Description
RECOVERY OFF	Turns off the Event Log. Nothing is displayed.
RECOVERY ERROR	Displays only errors.
RECOVERY ON	Displays all events.

Sample entries in the Event Log include the following:

- **Time.** The date and time the event occurred.
- **Server User.** The WebFOCUS Reporting Server user ID.
- **E-mail Address.** The user's e-mail address.
- **Message.** Description of the error that occurred.

Procedure

How to Monitor E-mail Traffic Using the Event Log

1. From the Two-Way Email Administrator Console, click the Event Log tab.

Note: The following screen displays if you have installed only Self-Service Two-Way Email.

The screenshot shows the Two-Way Email Administrator console. The top header includes the WebFOCUS logo and the text "Two-WayEmail". The main content area is titled "Two-Way Email Administrator" and contains two sections: "Event Log Report" and "Purge Event Log".

Event Log Report Section:

- Start Date: Oct 1 2001
- WebFOCUS Server User: [Text Input Field]
- Email Address: [Text Input Field]
- Click to View [Button]

Purge Event Log Section:

- Ending Date: Oct 9 2001
- Click to Purge [Button]

The left sidebar contains navigation links: Home, Two Way Email: Create a Template, Console, and Information Builders. Under Information Builders, there are three tabs: Job Log, Event Log (selected), and Job Status.

2. Apply selection criteria to define the information that will be retrieved:
 - a. Use the drop-down lists for Start Date to specify the beginning of the timeframe that you are interested in. Information on all events that occurred from that date up to the current date will be retrieved.
 - b. Type a WebFOCUS Reporting Server user ID in the WebFOCUS Server User field. Information only for the events associated with that user ID during the specified timeframe will be retrieved.
 - c. Type an e-mail address in the Email Address field. Information only for the events associated with the supplied user ID and address during the specified timeframe will be retrieved.
3. Click Click to View to display the Event Log report.

Purging the Event Log

Since the Event Log accumulates information and can become difficult to navigate, it is a good idea to periodically purge it to conserve space.

Procedure

How to Purge the Event Log

1. From the Two-Way Email Administrator Console, click the Event Log tab.
2. Using the drop-down lists for Ending Date, specify a date through which the Event Log will be purged.
3. Click Click to Purge. Respond to the confirmation message on the next dialog box. Self-Service Two-Way Email will delete all reports from the beginning of the Event Log through the specified ending date.

Checking the Status of a Job or Canceling a Job

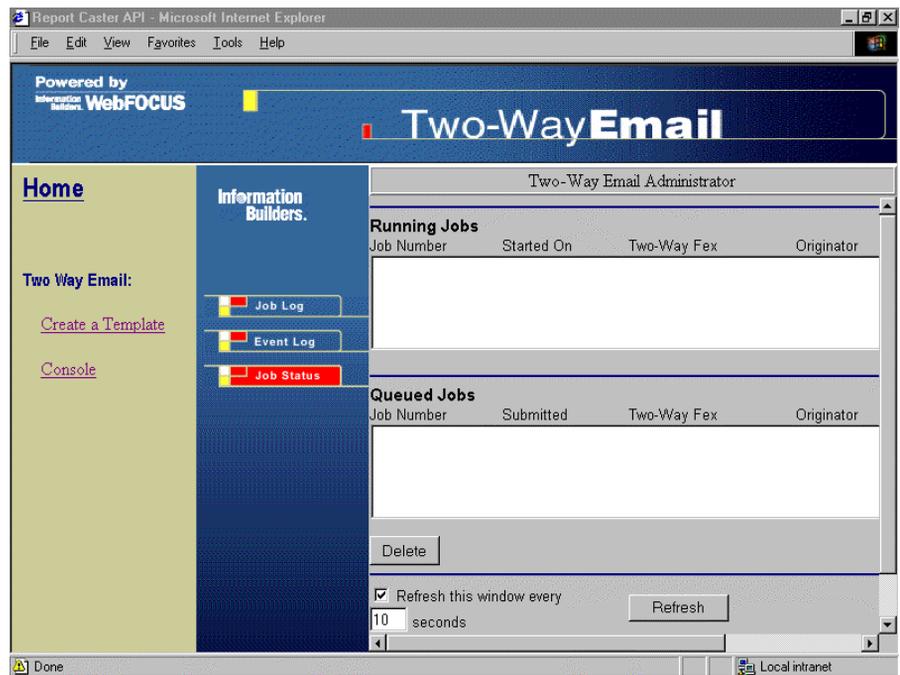
You can check the status of an executing job or a job that is waiting to be sent to the WebFOCUS Reporting Server for execution (a queued job). You can also cancel a queued job.

Procedure

How to Check the Status of a Job

1. From the Two-Way Email Administrator Console, click the Job Status tab.

Note: The following screen displays if you have installed only Self-Service Two-Way Email.



Jobs currently running on the WebFOCUS Reporting Server are displayed at the top of the window, under Running Jobs.

Jobs waiting to be sent to the WebFOCUS Reporting Server for execution are displayed in the middle of the window, under Queued Jobs.

The following information identifies a job:

- **Job Number.** The unique, system-generated key for a specific execution of a report.
 - **Started On** (under Running Jobs). The date and time the job started running on the WebFOCUS Reporting Server.
 - **Submitted** (under Queued Jobs). The date and time the job was submitted to the WebFOCUS Reporting Server for execution.
 - **Two-Way Fex.** The name of the procedure.
 - **Originator.** The e-mail address from which the request originated.
2. The Refresh this window every n seconds option is checked by default, retrieving the latest information after the specified interval of time. You can accept the default interval of 10, or supply the interval in seconds. Click Refresh to immediately retrieve the latest information.

Procedure

How to Cancel a Queued Job

Select one or more jobs from the list under Queued Jobs. Click Delete to cancel execution.

ReportCaster API Debugging Techniques

Topics:

- Tracing
- WebFOCUS Reporting Server Tracing
- Servlet Tracing
- Distribution Server Tracing
- Examining a Query String
- Using the API Query Tool

You can use several tools and techniques to debug your ReportCaster API application:

- Use trace files to monitor processes on the WebFOCUS Reporting Server, servlets, and Distribution Server.
- Examine a query string passed to a servlet after submitting a form. The query string may reveal errors you will want to fix. This technique applies only to direct HTML calls.
- Use the API Query Tool to quickly test a procedure that calls an API subroutine. This tool applies only to procedure-based calls.

Tracing

Trace ReportCaster API actions using the standard ReportCaster tracing facilities. All servlets and subroutines can be traced using servlet tracing.

The DSTLOG servlet provides a high-level summary about the execution of a job. For more information, see Chapter 3, *ReportCaster Servlet API*. However, the DSTRUNNOW servlet and DSTRUN subroutine interact with the Distribution Server and WebFOCUS Reporting Server. Therefore, they can be traced using the tracing facilities of those components. For a more detailed, system-level summary, use the tracing facilities of the WebFOCUS Reporting Server and Distribution Server.

WebFOCUS Reporting Server Tracing

WebFOCUS Reporting Server tracing provides information about job execution and output distribution for jobs executed with DSTRUNNOW and DSTRUN:

- SET TRACEUSER=ON sets tracing on during connectivity.
- SET TRACEUSER=OFF sets tracing off during connectivity.

These SET statements can be included in the WebFOCUS Reporting Server profile (edasprof), a user profile, or a procedure.

IBITRACE Facility

In addition to the SET TRACEUSER=ON command, you must add a SET command to the WebFOCUS Reporting Server trace file, IBITRACE.FEX. This file is read during the initialization of a server and contains:

- Information about the internal components to trace.
- A trace level for each component.
- A file name that determines the destination of the trace output for each component.
- Commands that enable a trace stamp, specify its location in the trace line, and indicate where line wrapping occurs.

Syntax**How to Customize the IBITRACE Facility**

To customize the trace facility, IBITRACE should contain one or more SET TRACE commands using the following format:

```
SET {TRACEON|TRACEOFF} = component[/level[/filename]]
```

or

```
SET TRACEON=ALL
```

Enables all traces.

```
SET TRACEOFF=ALL
```

Disables all traces.

where:

TRACEON

Enables tracing for a specified internal component.

TRACEOFF

Disables tracing for a specified internal component.

component

Specifies an internal component for tracing. Specific ReportCaster values include:

- **SNDH/1**—writes all communication information to the WebFOCUS Reporting Server trace files associated with the job's execution process.
- **SNDH/2**—writes all communication information to the WebFOCUS Reporting Server trace files associated with the job's execution process. In addition, SNDH/2 writes the following to the WebFOCUS Reporting Server trace files:
 - Distribution List and SEND command information.
 - ReportCaster subroutine information.

level

Specifies a trace level for the internal component. When the value is 0, the trace level is off. When the value is 1, the trace level is on.

filename

Specifies the file name to which the trace output is sent. Each traceable internal component has a default file name to which it sends output.

For more information about valid component values, valid trace levels, and default file names, see the iWay Server documentation for your platform.

Servlet Tracing

Servlet tracing is enabled with `SERVLETTRACE ON` in the servlet section of the Distribution Server configuration file (`/ibi/distributionserver436/cfg/bkrsched.cfg`). It enables tracing for all servlets and subroutines executed by the ReportCaster API.

Servlet tracing provides information about queries to and maintenance of the ReportCaster Repository, including events that occur during the scheduling of a job or while you are running the console. Tracing messages are sent to servlet trace files managed by the Web server, as well as to the file specified by the `SRVTRACEFILE` parameter in the Distribution Server configuration file.

Distribution Server Tracing

A `schedule.log` file is always created in the `/log` directory under the Distribution Server home directory (`/ibi/distributionserver436/`). This file traces Distribution Server initialization and indicates the options turned on with the Distribution Server configuration file.

Additional Distribution Server tracing is enabled with the parameter `SCHTRACE ON` in the `SCHEDULE` section of the `bkrsched.cfg` configuration file. `SCHTRACE ON` creates trace files in the `/trc` directory under the Distribution Server home directory. When a new instance of the Distribution Server starts, current log and trace files are appended to previous log and trace files.

Distribution Server Tracing Files

The following files contain information when you set `SCHTRACE ON` in `bkrsched.cfg`:

- **main.trc** traces the commands processed by the main Distribution Server thread. These commands include initialization, shutdown, and running a job immediately.
- **reader.trc** traces the Distribution Server processing that takes place every minute.
- **disp.trc** traces the date and time communication threads that open and close between the Distribution Server and the WebFOCUS Reporting Server. The number of threads is defined by the `MAX_THREADS` parameter in the `bkrsched.cfg` file. The default number of `MAX_THREADS` is 3.
- **console.trc** traces all communication from the servlets and Java applications to the Distribution Server. This communication takes place at various points during ReportCaster processing and is traced when:
 - WebFOCUS (Windows version) stores a scheduled procedure on the Distribution Server.
 - The `SCHStop` program is run to stop the Distribution Server.

Trace Files for Individual Jobs

Setting SCHTRACE ON in the SCHEDULE section of the bkrsched.cfg configuration file causes trace files for individual jobs to be created in the /trc directory under the Distribution Server home directory. Identify the process ID of the target job execution by running a log report from one of the following:

- ReportCaster Interface
- ReportCaster Console
- DSTLOG API servlet

Each job creates two trace files (each with a different suffix), and is named with a unique ID assigned to the job. The job ID is a base 32 number beginning with P, followed by a random series of 10 digits and lowercase letters, such as P0rlpimsb14 or P0rlpj3o932.

Each job creates two trace files with a unique job ID. The job ID begins with a P and is followed by a random series of 10 digits and lower case letters, such as P0rlpimsb14. The two trace files that are created are:

- **.trc** is the job trace file that contains all information related to the execution of a job, including the scheduled procedure, distribution information, and log creation and contents.
- **.fex** is dynamically generated by the Distribution Server and submitted to the WebFOCUS Reporting Server. This procedure contains:
 - Distribution information.
 - Report parameters (if any) stored with the schedule.
 - Pre-execution and post-execution steps (if any) for the scheduled procedure.
 - Procedure code (if MRE or Windows version) or a -INCLUDE FOCEXEC (if a server-based procedure).

At the end of each job run there are two files in the /trc directory. For example, for the job ID of P0rlpimsb14, P0rlpimsb14.trc and P0rlpimsb14.fex reside in the /trc directory.

Tracing Error Files

When ReportCaster encounters an unexpected error or abend, the following tracing error files are created:

- **console.err** when the console terminates.
- **disp.err** when the dispatcher terminates.
- **main.err** when the main thread terminates.
- **reader.err** when the reader terminates.
- **P*.err** when a thread running a scheduled job ID terminates. The name of the file is the same as the scheduled job's ID. For example, if a scheduled job with a P0rlpj3o932 job ID encounters an error, a P0rlpj3o932.err job error file is created.

Examining a Query String

When you code an HTML form that directly calls an API servlet, you can examine the query string generated by the form after it is submitted. The query string may reveal omissions or errors that require correction.

Procedure

How to Examine a Query String

1. In the HTML code for the form, change the METHOD from POST to GET. For example:

```
<FORM ACTION="/servlet/DSTDLBULK" METHOD="GET" ENCTYPE="application/x-www-form-urlencoded">
```
2. Enter values in the fields on the form.
3. Submit the form.
4. Copy the generated query string that appears on the address line and paste it to another file, using any standard text editor.
5. Examine the query string, making sure that it includes all required parameters for the servlet that is called.

Example**Examining a Query String for the DSTDLBULK Servlet**

This example illustrates a valid query string generated by the sample HTML calling form `rbabulkm.htm`, which runs the DSTDLBULK servlet. For more information about the `rbabulkm.htm` form, see Chapter 6, *ReportCaster API Servlet Samples*.

The `rbabulkm.htm` form enables a user to create a new Distribution List by entering address names in a text box. A form that calls the DSTDLBULK servlet to create a new Distribution List from text box input must generate a query string containing the following required parameters:

```
IBIB_name
IBIB_access
IBIB_method
IBIB_function
IBIB_recipients
```

Assume that you are testing this form for use in your application. Enter MYLIST1 as the name (IBIC_name) of the new Distribution List. Accept the default values for access authorization (IBIC_access), distribution method (IBIC_method), and function (IBIC_function). In the IBIC_recipients text box, enter an individual e-mail address such as:

```
Alfred_Stevens@abcd.com,,$
```

When you submit the form, the query string correctly appears on the address line as:

```
http://hostname/rcaster/servlet/DSTDLBULK?IBIB_name=MYLIST1owner=
access=PUMethod=MAILfunction=C
recipients=Alfred_Stevens@abcd.com%2C%2C%24
submit=Submit
```

where:

```
hostname
```

Is the host name of the Web server.

Note: The characters `%2C` are substitutions that the browser makes for HTML reserved characters such as commas.

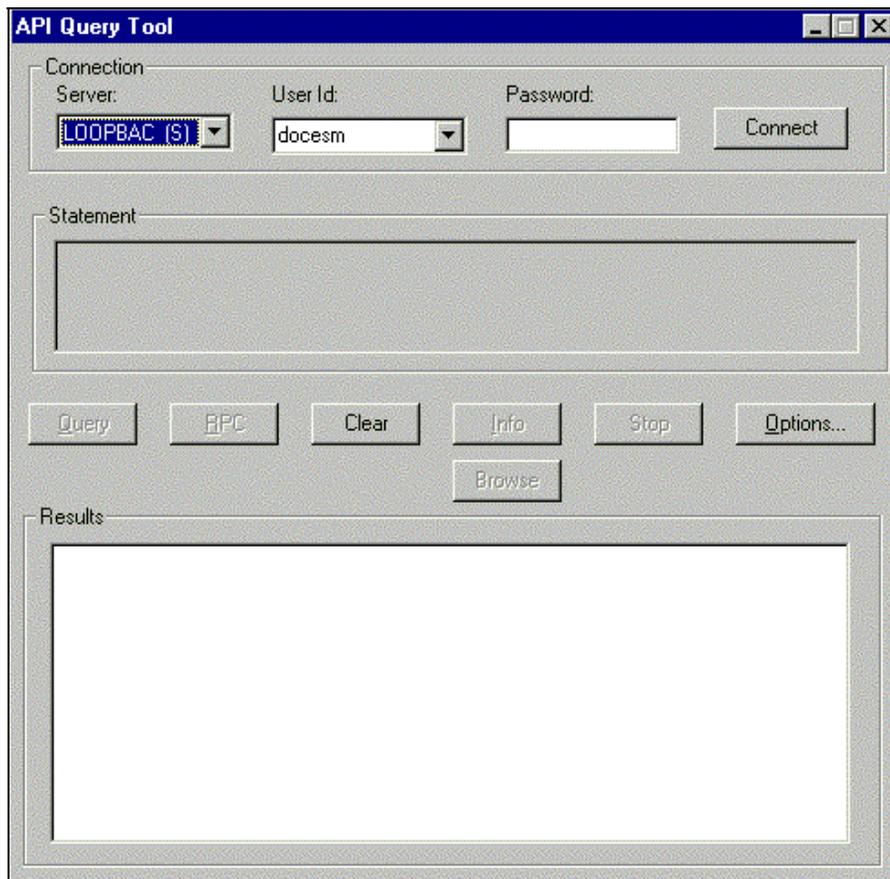
Using the API Query Tool

On Windows NT, the API Query Tool can be used to troubleshoot procedures that call ReportCaster API subroutines. This tool enables you to quickly determine if a procedure is properly coded and returns the expected results.

Procedure

How to Use the API Query Tool

1. From the Programs menu, select WebFOCUS 436 Server and then click API Query (Installation Verification) Tool. A dialog box similar to the following displays:



2. Select the name of the WebFOCUS Reporting Server and the user ID from the drop-down list. Enter your password.
3. Click **Connect** to establish communications with the WebFOCUS Reporting Server.
4. Once you have connected successfully, the **Statement** field box is enabled.
5. In the **Statement** field box, enter the name of the procedure that calls the **ReportCaster** subroutine.
6. Click the **RPC** button.

The results of the procedure display in the **Results** window.

7. Check the log report in **ReportCaster** to verify that the job ran.
8. You can also check the applicable trace file for messages from the subroutine that was called. Subroutine tracing is enabled with the command **SERVLETTRACE ON** in the **servlet** section of the **ReportCaster** configuration file (**bkrsched.cfg**). Messages are written to the path and file specified on the **SRVTRACEFILE** parameter in the configuration file (for example, **c:\servlet.trc**).

For details about the **ReportCaster** configuration file, see the *WebFOCUS ReportCaster Administrator's Manual*.

APPENDIX B

ReportCaster API Messages and Codes

Topic:

- ReportCaster API Subroutine Messages
- ReportCaster Bean API Messages
- ReportCaster Servlet API Messages

The ReportCaster API returns messages and codes that indicate whether or not a bean, servlet, or subroutine completed successfully.

ReportCaster API messages and codes are not added to the ReportCaster log file.

ReportCaster API Subroutine Messages

A ReportCaster API subroutine called by a procedure provides a return code that indicates whether or not a subroutine completed successfully.

The last argument passed to a ReportCaster API subroutine is always the format of the return code ('I4'). The value of the return code is supplied by the subroutine upon execution, and returned to the calling procedure so that the user may perform error handling. A return code of zero (0) means that the subroutine completed successfully.

ReportCaster supplies three procedures that translate return codes into messages. Each procedure is associated with a specific ReportCaster API subroutine. The procedures provide the user with a helpful message. For example, the return code -3 is translated into the message "The password is not valid."

On Windows NT/2000, the following translation procedures are located in EDAHOME\catalog:

- DSTRUNER for the DSTRUN subroutine.
- DSTMEMER for the DSTMEM subroutine.
- DSTBLKER for the DSTBULK subroutine.

Messages can be written to a log file using a FILEDEF command to establish a permanent file location. For more information about the FILEDEF command, see your *WebFOCUS Developing Reporting Applications* manual.

DSTRUNER (DSTRUN Return Codes)

The following is the translation procedure DSTRUNER, which includes all return codes and messages for the DSTRUN subroutine.

```

-SET &ERRMSG=DECODE &SUBERR(
- 1      '1 - This is not a valid host'
- 2      '2 - The userid is not valid'
- 3      '3 - The password is not valid'
- 4      '4 - VXLOAD error'
- 5      '5 - TCPH CRT error'
- 6      '6 - HTTP CRT error'
- 7      '7 - HTTP open error'
- 8      '8 - HTTP post error'
- 9      '9 - HTTP read error'
- 10     '10 - Argument Error'
- 11     '11 - Servlet Execution Error'
- 101    '101 - Failed to connect to the Distribution Server.'
- 102    '102 - No WebFOCUS user logged on.'
- 103    '103 - The schedule ID is missing.'
- 104    '104 - A Priority has not been specified.'
- 105    '105 - Configuration error.'
- 106    '106 - Unknown action.'
- 107    '107 - Necessary field(s) not defined.'
- 108    '108 - The Distribution Server is unavailable.'
- 109    '109 - Unable to connect to the database.'
- 110    '110 - There are no pending Job(s).'
- 111    '111 - System error.'
- 112    '112 - Specify EITHER Job Description or Schedule Id - NOT both.'
- 113    '113 - IBIB priority must be a number between 1-5.'
- 114    '114 - Error sending run immediate request to scheduler.'
- 116    '116 - Job scheduled for execution.'
- 117    '117 - Unable to queue job for execution.'
- 118    '118 - You must log in to the remote console.'
- 119    '119 - The schedule ID is not in the Queue.'
- 120    '120 - The schedule ID must be entered.'
- 121    '121 - The schedule priority must be entered.'
- 122    '122 - Priority is not correct. Select a number between 1 and 5.'
- 123    '123 - The requested action is unrecognized.'
- 124    '124 - No output from remote console. Is Dist.Server active?'
- 125    '125 - The field has an error.'
- 126    '126 - Internal Error.'
- 127    '127 - No Query Parameter.'
- 128    '128 - User not authorized for query. No cookie information.'
- 129    '129 - Distribution list not specified.'
- 130    '130 - Value is missing.'
- 131    '131 - IBI_user is missing.'
- 132    '132 - Destination is missing.'
- 133    '133 - Function undefined.'
- 134    '134 - Sql method error.'
- 135    '135 - Userid undefined.'
- 136    '136 - Unable to update the database.'

```

```
- 137 '137 - Insufficient privileges.'  
- 138 '138 - Address already exists in the target list.'  
- 139 '139 - Address does not exist in the target list.'  
- 140 '140 - Distribution list owner not defined.'  
- 141 '141 - Distribution list not specified.'  
- 142 '142 - Distribution method undefined.'  
- 143 '143 - Function not specified.'  
- 144 '144 - Access (public/private) not defined.'  
- 145 '145 - List of recipients not provided.'  
- 146 '146 - Enter either file or recipient information (not both)'  
- 147 '147 - File does not exist or the data is incorrect.'  
- 148 '148 - Distribution data error.'  
- 149 '149 - No data in the file.'  
- 150 '150 - The distribution list is not in the database.'  
- 151 '151 - Sql method error.'  
- 152 '152 - Userid undefined.'  
- 153 '153 - IBIB_user must be specified.'  
- 154 '154 - Failed to read the config file ibidir.wfs from WebFOCUS.'  
- 155 '155 - Failed to read host name of Dist. Server from WebFOCUS.'  
- 156 '156 - Failed to read port number of Dist. Server from WebFOCUS.'  
- 157 '157 - Failed to test JDBC and WebFOCUS Server connections.'  
- 158 '158 - Failed to read the resource files from WebFOCUS.'  
- 159 '159 - Failed during Job Description verification.'  
- 160 '160 - Unable to save schedule. Job Description already exists.'  
- 161 '161 - Unable to update schedule. Another user has deleted it.'  
- 162 '162 - Unable to update address book. Another user has deleted it.'  
- 163 '163 - User specified is not in the database.'  
- 164 '164 - Unable to update log report. Another user has deleted it.'  
- 165 '165 - Unable to delete schedule. Another user has deleted it.'  
- 166 '166 - Unable to delete address book. Another user has deleted it.'  
- 167 '167 - Unable to delete user. The user is not in the database.'  
- 168 '168 - Unable to delete log report. Another user has deleted it.'  
- 169 '169 - Unable to save schedule. Another user has created it.'  
- 170 '170 - Unable to save address book. Another user has created it.'  
- 171 '171 - The user already exists in the database.'  
- 172 '172 - Unable to save log report. Another user has created it.'  
- 173 '173 - Unable to open schedule. Another user has deleted it.'  
- 174 '174 - Unable to open address book. Another user has deleted it.'  
- 175 '175 - Unable to open user. Another user has deleted it.'  
- 176 '176 - Unable to open log report. Another user has deleted it.'  
- 177 '177 - Field limits and character limits are violated.'  
- 178 '178 - Invalid Start Date.'  
- 179 '179 - Invalid Start Time.'  
- 180 '180 - Invalid End Date.'  
- 181 '181 - Invalid End Time.'  
- 182 '182 - Schedule id must be entered.'  
- 183 '183 - The schedule does not exist for the specified user.'  
- 184 '184 - Unable to connect to ReportCaster Repository Server.'  
- 185 '185 - Unable to connect to WebFOCUS Reporting Server.'  
- 186 '186 - Record has already been added.'  
- 187 '187 - Record has already been deleted.'  
- 188 '188 - Record has already been updated. Resubmit transaction.'  
- ELSE ' '');
```

DSTMEMER (DSTMEM Return Codes)

The translation procedure DSTMEMER includes all return codes and messages for the DSTMEM subroutine. These return codes are exactly the same as for DSTRUN, except that the codes are in the 200 range instead of 100. For example:

The following is a return code for DSTRUN

```
- 137 '137 - Insufficient privileges.'
```

The following is a return code for DSTMEM:

```
- 237 '237 - Insufficient privileges.'
```

DSTBLKER (DSTBULK Return Codes)

The translation procedure DSTBLKER includes all return codes and messages for the DSTBULK subroutine. These return codes are exactly the same as for DSTRUN, except that the codes are in the 300 range instead of 100. For example:

The following is a return code for DSTRUN

```
- 137 '137 - Insufficient privileges.'
```

The following is a return code for DSTBULK:

```
- 337 '337 - Insufficient privileges.'
```

ReportCaster Bean API Messages

If properties have not been set for a function (or method) used in a JSP, a message will be returned when it is run. The following message is generated when you attempt to run the `rcaster_create.jsp` without setting its required properties:

```
IBIB_method can not be found.
```

ReportCaster Servlet API Messages

If parameters have not been supplied for a servlet, a message will be returned when it is run. The following message is generated when you attempt to run the `DSTRUNNOW` servlet without specifying a job description or schedule ID:

```
Either IBIB_jobdesc or IBIB_scheduleid must be specified.
```

APPENDIX C

API-based MRE User

Topics:

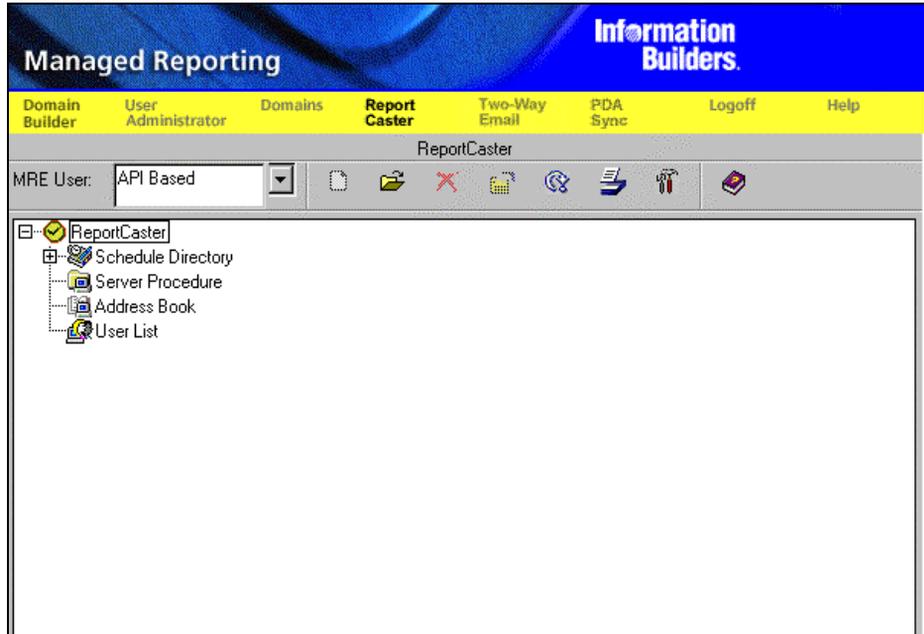
- Accessing API-based Schedules and Log Reports
- Maintaining a Server Schedule
- Viewing an API-based Log File

If you are a ReportCaster Administrator and you access ReportCaster from the yellow toolbar in the Managed Reporting Environment (MRE), you can select API Based as an MRE User type. This enables you to:

- Maintain server schedules created using the ReportCaster API.
- View log reports for jobs created using the ReportCaster API.

Accessing API-based Schedules and Log Reports

Upon logging into MRE, click ReportCaster from the yellow toolbar. To access API-based server schedules and log reports, select API Based from the MRE User drop-down list.



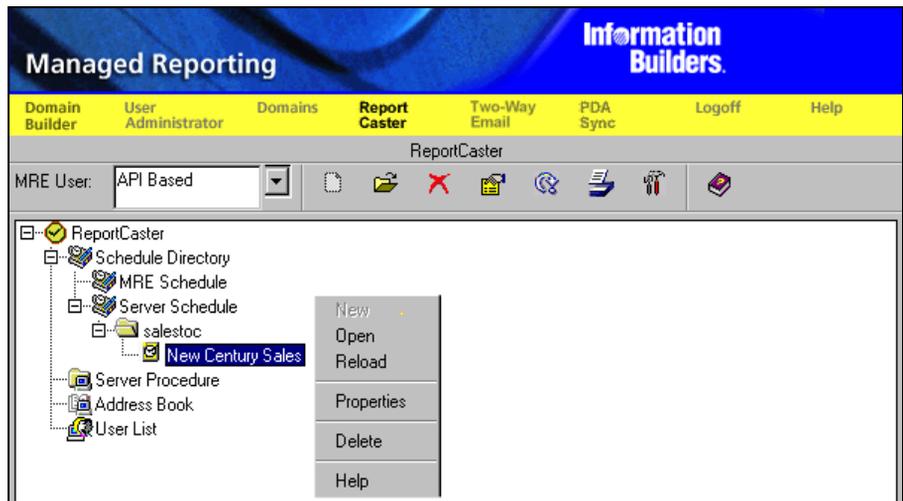
Maintaining a Server Schedule

The Server Schedule component enables you to access WebFOCUS Reporting Server schedules that were created using the ReportCaster API, provided that you access ReportCaster from the yellow toolbar in MRE.

Procedure

How to Access API-Based Server Schedules

1. Select the Schedule Directory component.
2. Select the Server Schedule component. The available server schedule folders display. These folders reflect the name of the server procedure (for example, salestoc).
3. Select a schedule (for example, New Century Sales). This schedule name reflects the job description that was specified at schedule time.
4. Right-click on the schedule and select an available option. These options can also be accessed using the icons on the gray toolbar.



Server Schedule Options

Once you have selected an API-based server schedule, the following options are available:

- **Open.** Enables you to edit an API-based schedule. For more information about scheduling a procedure from MRE, see the *WebFOCUS ReportCaster Administrator's Manual*.
- **Reload.** Retrieves information from the ReportCaster Repository and refreshes the display with current information. Any open scheduled job will be closed. The Schedule Directory component will remain open, and its content will accurately reflect repository information.
- **Properties.** Displays the properties for an API-based schedule. This option enables you to change the WebFOCUS Reporting Server user ID that is associated with the schedule.
- **Delete.** Removes your selection from the display and deletes the scheduled job from the ReportCaster Repository. This option does not delete the folder in the Managed Reporting Repository.
- **Help.** Accesses the ReportCaster online help file. The help information displays in a new browser window. To return to the ReportCaster Interface, close the browser window displaying the help information.

Viewing an API-based Log File

The Log File icon  on the gray toolbar enables you to generate log reports for schedules that were created using the ReportCaster API, provided that you access ReportCaster from the yellow toolbar in MRE.

When you select the Log File icon, the Log Options screen displays in a separate window, enabling you to select options (job description, start date, and start time) to generate a log report for API-based jobs.

You can also view a Log File by clicking the Console icon  to access the ReportCaster Console.

For more information about selecting Log File options in MRE, see the *WebFOCUS ReportCaster Administrator's Manual*.

Index

A

- absolute addresses, 3-8
 - specifying, 3-10, 3-17
- access argument for DSTBULK, 4-11
- access_length argument for DSTBULK, 4-11
- Address Book reports, 7-1, 7-7
 - running, 7-2
- Administrator console for Two-Way Email, 8-9, 10-1 to 10-2
 - canceling jobs, 10-7
 - Event Log, 10-5, 10-7
 - Job Log, 10-2 to 10-4
 - monitoring e-mail traffic, 10-5
 - viewing job status, 10-7
- Alert Schedules reports, 7-1, 7-9
 - running, 7-2
- amper variables for subroutines, 4-34
- API Query Tool, A-1, A-8
- API-based MRE user IDs, C-1 to C-2
- application servers, 1-2

B

- Beans components, 1-2
- BOTADDR table, 3-4, 3-10, 3-19, 4-2, 4-8
- BOTDEST table, 3-2, 3-4 to 3-5, 3-10, 3-17, 3-19, 4-2, 4-8, 4-18
- BOTEUSER table, 2-4 to 2-5, 3-4 to 3-5, 4-2
- BOTLOG table, 2-4 to 2-5, 3-4 to 3-5
- BOTLOG2 table, 2-4 to 2-5, 3-4 to 3-5

- BOTPARMS table, 2-4 to 2-5, 3-4 to 3-5, 4-2
- BOTSCHED table, 2-4 to 2-5, 3-4 to 3-5, 3-20, 4-2, C-3 to C-4
- burst reports, 3-15
 - creating an external file, 3-15
 - generating text box input, 3-16

C

- calling a servlet from an HTML form, 3-8
- canceling Two-Way Email jobs, 10-7
- Change Status of Scheduled Job window, 6-12
- classes, 1-2
- coding requirements for creating an external file, 3-15
- Conditional Schedules reports, 5-16 to 5-17
- console.err tracing files, A-6
- console.trc files, A-4
- copy argument for DSTBULK, 4-14
- Copy Distribution Lists window, 6-18
- copy_length argument for DSTBULK, 4-14
- Create a New Schedule window, 5-7, 6-6
- Create/Update Distribution Lists from a file window, 6-14
- creating a Self-Service Two-Way Email template, 9-4
- creating an external file, 3-15 to 3-16
- creating schedules using a JSP, 5-7
- creating schedules using a servlet, 6-5 to 6-6

D

- Data Type Definitions (DTDs), 2-12
 - ReportCaster Log List, 2-16
 - ReportCaster Log Process List, 2-17
 - ReportCaster Process Log, 2-17
 - Schedule Detail, 2-14
 - Schedule List, 2-12
 - Schedule Properties, 2-13
- debugging ReportCaster API, A-1
- destfn argument for DSTMEM, 4-22
- destfn_length argument for DSTMEM, 4-22
- Dialogue Manager procedures, 4-4, 4-15, 4-24
- direct HTML calls, 2-3, 3-3, 3-10
 - DSTACTIVE, 3-35
 - DSTDLLIST, 3-19
 - DSTRUNNOW, 3-37
 - DSTSCHEM, 3-20
- disp.err tracing files, A-6
- disp.trc files, A-4
- Display Distribution Lists window, 6-19
- Display Schedules with Conditions window, 5-17
- distribution lists, 3-1 to 3-2, 3-10
 - adding recipients, 3-10, 3-17, 4-8, 4-18, 4-24, 6-14
 - copying, 3-10, 4-8, 6-14, 6-18
 - creating, 3-10, 4-1, 4-8, 6-14
 - deleting, 3-10, 4-8, 6-14
 - deleting recipients, 3-17, 4-18
 - displaying, 6-19
 - maintaining, 3-10, 4-1 to 4-2, 4-8, 4-18, 6-14 to 6-15
 - replacing recipients, 3-10, 4-8, 4-15
 - updating, 6-14
- Distribution Server tracing, A-4 to A-6
- Distribution Type reports, 7-1, 7-5
 - running, 7-2, 7-5
- DSTACTIVE servlet, 3-2, 3-4 to 3-5, 3-7, 3-35, 6-12
 - requirements, 3-35
- DSTACTIVE servlet (*continued*)
 - setting status of reports, 3-35
 - specifying a relative address, 3-35
- DSTACTIVE servlet parameters, 3-35
 - IBIB_active, 3-35
 - IBIB_jobdesc, 3-36
 - IBIB_scheduleid, 3-36
 - IBIB_tcpiplevel, 3-36
- DSTAPIStatus Bean, 2-2, 8-2 to 8-3
- DSTAuthenticate Bean, 2-2, 2-6, 8-2 to 8-3
 - properties, 2-7
- DSTBeanHandler Bean, 2-2, 8-2 to 8-3
- DSTBeanResult Bean, 2-2, 8-2 to 8-3
- DSTBLKER translation procedure, B-2, B-5
- DSTBULK subroutine, 4-2, 4-8, 4-15, B-2
 - maintaining multiple members, 4-2
 - return codes, B-5
- DSTBULK subroutine arguments, 4-8
 - access, 4-11
 - access_length, 4-11
 - copy, 4-14
 - copy_length, 4-14
 - filename, 4-13
 - filename_length, 4-13
 - function, 4-11
 - function_length, 4-11
 - hostname_port, 4-10
 - hostname_port_length, 4-10
 - httpuser/pswd, 4-14
 - httpuser/pswd_length, 4-14
 - method, 4-12
 - method_length, 4-12
 - name, 4-10
 - name_length, 4-10
 - returncode, 4-15
 - srv_userid, 4-9
 - srv_userid_length, 4-9
 - srv_userpass, 4-9
 - srv_userpass_length, 4-9
 - tcpiplevel, 4-15
 - user, 4-12
 - user_length, 4-13

- DSTDLBULK servlet, 3-2, 3-4 to 3-5, 3-7, 3-10, 6-14 to 6-15, 6-18, A-7
 - specifying an absolute address, 3-10
- DSTDLBULK servlet parameters, 3-11
 - IBIB_access, 3-11
 - IBIB_copy, 3-11
 - IBIB_filename, 3-12
 - IBIB_function, 3-12
 - IBIB_method, 3-13
 - IBIB_name, 3-13
 - IBIB_recipients, 3-13
 - IBIB_tcpiplevel, 3-14
 - IBIB_userid, 3-14
- DSTDLLIST servlet, 3-2, 3-4 to 3-5, 3-7, 3-10, 3-19, 6-19
 - displaying distribution lists, 3-19
 - specifying a relative address, 3-19
- DSTDLLIST servlet parameters, 3-19
 - IBIB_name, 3-20
 - IBIB_tcpiplevel, 3-20
- DSTDLMEM servlet, 3-2, 3-4 to 3-5, 3-7, 3-10, 3-17, 6-17
 - specifying an absolute address, 3-17
- DSTDLMEM servlet parameters, 3-17
 - IBIB_function, 3-17
 - IBIB_location, 3-18
 - IBIB_name, 3-18
 - IBIB_tcpiplevel, 3-18
 - IBIB_userid, 3-19
 - IBIB_value, 3-19
- DSTLOG servlet, 3-2, 3-4 to 3-5, 3-7, 3-40, 6-20
 - specifying a relative address, 3-40
- DSTLOG servlet parameters, 3-40
 - IBIB_enddate, 3-40
 - IBIB_endtime, 3-41
 - IBIB_jobdesc, 3-41
 - IBIB_lastexec, 3-41
 - IBIB_scheduleid, 3-42
 - IBIB_startdate, 3-42
 - IBIB_starttime, 3-42
 - IBIB_tcpiplevel, 3-43
- DSTLogFactory Bean, 2-2, 2-4
- DSTMEM subroutine, 4-2, 4-18, 4-24, B-2
 - return codes, B-5
- DSTMEM subroutine arguments, 4-18
 - destfn, 4-22
 - destfn_length, 4-22
 - fldvlu, 4-21
 - fldvlu_length, 4-22
 - function, 4-20
 - function_length, 4-21
 - hostname_port, 4-19
 - hostname_port_length, 4-20
 - httpuser/pswd, 4-22
 - httpuser/pswd_length, 4-23
 - name, 4-20
 - name_length, 4-20
 - owner, 4-21
 - returncode, 4-23
 - srv_userid, 4-18
 - srv_userid_length, 4-19
 - srv_userpass, 4-19
 - srv_userpass_length, 4-19
 - tcpiplevel, 4-23
 - user_length, 4-21
- DSTMEMER translation procedure, B-2, B-5
- DSTPeriodSchedule Bean, 2-2
- DSTRUN subroutine, 4-25 to 4-26, 4-32, B-2
 - arguments, 4-26
 - requirements, 4-26
 - return codes, B-3
 - running a report immediately, 4-26, 4-32
- DSTRUN subroutine arguments, 4-26
 - hostname_port, 4-28
 - hostname_port_length, 4-28
 - httpuser/pswd, 4-31
 - httpuser/pswd_length, 4-31
 - jobdesc, 4-29
 - jobdesc_length, 4-29
 - owner, 4-30
 - PARM, 4-30
 - parm_length, 4-30
 - priority, 4-29
 - returncode, 4-31
 - scheduleid, 4-28
 - scheduleid_length, 4-29
 - srv_userid, 4-27

DSTRUN subroutine arguments (*continued*)

- srv_userid_length, 4-27
- srv_userpass, 4-27
- srv_userpass_length, 4-27
- tcpiplevel, 4-31
- user_length, 4-30

DSTRUNER translation procedure, B-2 to B-3

DSTRUNNOW servlet, 3-2, 3-4 to 3-5, 3-7, 3-37, 6-13

- requirements, 3-37
- running a report immediately, 3-37
- specifying a relative address, 3-37

DSTRUNNOW servlet parameters, 3-38

- IBIB_jobdesc, 3-38
- IBIB_parm, 3-38
- IBIB_priority, 3-38
- IBIB_scheduleid, 3-39
- IBIB_tcpiplevel, 3-39
- IBIB_userid, 3-39

DSTSCHED servlet, 3-2, 3-4 to 3-5, 3-7, 6-6

- passing values, 3-33 to 3-34
- requirements, 3-20
- scheduling a job, 3-20
- specifying a relative address, 3-21

DSTSCHED servlet parameters, 3-21

- IBIB_active, 3-21
- IBIB_asvalue, 3-30
- IBIB_byfield, 3-21
- IBIB_dates, 3-22
- IBIB_distlist, 3-22
- IBIB_enddate, 3-22
- IBIB_endtime, 3-23
- IBIB_frequency, 3-23
- IBIB_ftphost, 3-30
- IBIB_ftplocation, 3-30
- IBIB_ftppass, 3-31
- IBIB_ftpuser, 3-31
- IBIB_interval, 3-23
- IBIB_jobdesc, 3-24
- IBIB_jobname, 3-24
- IBIB_mailcompany, 3-28
- IBIB_mailfrom, 3-29
- IBIB_mailhost, 3-29
- IBIB_maisubject, 3-29
- IBIB_method, 3-24

DSTSCHED servlet parameters (*continued*)

- IBIB_notifyaddress, 3-31
- IBIB_notifybrief, 3-32
- IBIB_notifyflag, 3-25
- IBIB_notifyreply, 3-32
- IBIB_notifysubject, 3-32
- IBIB_parm, 3-25, 3-33
- IBIB_postrpc1, 3-25
- IBIB_postrpc2, 3-25
- IBIB_prerpc1, 3-26
- IBIB_prerpc2, 3-26
- IBIB_sendformat, 3-26
- IBIB_startdate, 3-26
- IBIB_starttime, 3-27
- IBIB_tcpiplevel, 3-27
- IBIB_weekdays, 3-28

DSTSchedFactory Bean, 2-2, 2-4 to 2-5

DSTSelectionHandler Bean, 2-2

DSTServlets jar file, 8-2

DSTTwoWayFactory Bean, 2-2, 8-2 to 8-3, 9-2, 9-5

DSTTWSelfHandler Bean, 8-2 to 8-3

DTDs (Data Type Definitions), 2-12

- ReportCaster Log List, 2-16
- ReportCaster Log Process List, 2-17
- ReportCaster Process Log, 2-17
- Schedule Detail, 2-14
- Schedule List, 2-12
- Schedule Properties, 2-13

E

EDAPATH variable, 3-2, 3-10, 3-12, 3-20, 4-2 to 4-3, 4-7 to 4-8, 4-13

EDARPC method call, 4-3

Event Log for Two-Way Email, 10-5

- purging, 10-6 to 10-7

examining a query string, A-1, A-6 to A-7

external files, 3-15 to 3-16

- F**
- filename argument for DSTBULK, 4-13
 - filename_length argument for DSTBULK, 4-13
 - fldvlu argument for DSTMEM, 4-21
 - fldvlu_length argument for DSTMEM, 4-22
 - FOCUS proprietary data sources, 2-3, 3-3 to 3-4, 4-2
 - FTP parameters for DSTSCHED, 3-30
 - function argument for DSTBULK, 4-11
 - function argument for DSTMEM, 4-20
 - function_length argument for DSTBULK, 4-11
 - function_length argument for DSTMEM, 4-21
- G**
- generating text box input, 3-16
 - get methods, 2-12
 - getTWSelfHandler method, 9-2, 9-5
 - gray toolbar icons, C-4
 - group folder description options, C-3 to C-4
- H**
- hostname_port argument for DSTBULK, 4-10
 - hostname_port argument for DSTMEM, 4-19
 - hostname_port argument for DSTRUN, 4-28
 - hostname_port_length argument for DSTBULK, 4-10
 - hostname_port_length argument for DSTMEM, 4-20
 - hostname_port_length argument for DSTRUN, 4-28
 - HTML calls, 2-3, 3-8, 3-10, 3-19
 - DSTACTIVE, 3-35
 - DSTDLLIST, 3-19
 - DSTRUNNOW, 3-37
 - HTML calls (*continued*)
 - DSTSCHED, 3-20
 - HTML forms, 3-2, 3-10, 3-16, 3-20, 4-2 to 4-3, 4-7 to 4-8
 - calling JavaServer Pages, 2-22 to 2-23
 - calling procedures, 4-7
 - HTML sample forms, 3-9, 6-1 to 6-2, 6-5
 - httpuser/pswd argument for DSTBULK, 4-14
 - httpuser/pswd argument for DSTMEM, 4-22
 - httpuser/pswd argument for DSTRUN, 4-31
 - httpuser/pswd_length argument for DSTBULK, 4-14
 - httpuser/pswd_length argument for DSTMEM, 4-23
 - httpuser/pswd_length argument for DSTRUN, 4-31
- I**
- ibi.broker.beans package, 2-2, 8-3
 - ibi.broker.beans.handler property, 2-2
 - IBIB_access parameter for DSTDLBULK servlet, 3-11
 - IBIB_active parameter for DSTACTIVE servlet, 3-35
 - IBIB_active parameter for DSTSCHED servlet, 3-21
 - IBIB_asvalue parameter for DSTSCHED servlet, 3-30
 - IBIB_byfield parameter for DSTSCHED servlet, 3-21
 - IBIB_copy parameter for DSTDLBULK servlet, 3-11
 - IBIB_dates parameter for DSTSCHED servlet, 3-22
 - IBIB_distlist parameter for DSTSCHED servlet, 3-22
 - IBIB_enddate parameter for DSTLOG servlet, 3-40
 - IBIB_enddate parameter for DSTSCHED servlet, 3-22

- IBIB_endtime parameter for DSTLOG servlet, 3-41
- IBIB_endtime parameter for DSTSCHED servlet, 3-23
- IBIB_fex property, 9-2, 9-5
- IBIB_filename parameter for DSTDLBULK, 3-12, 3-15
- IBIB_format property, 9-2, 9-5
- IBIB_frequency parameter for DSTSCHED servlet, 3-23
- IBIB_ftphost parameter for DSTSCHED servlet, 3-30
- IBIB_ftplocation parameter for DSTSCHED servlet, 3-30
- IBIB_ftppass parameter for DSTSCHED servlet, 3-31
- IBIB_ftpuser parameter for DSTSCHED servlet, 3-31
- IBIB_function parameter for DSTDLBULK servlet, 3-12, 3-15 to 3-16
- IBIB_function parameter for DSTDLMEM servlet, 3-17
- IBIB_interval parameter for DSTSCHED servlet, 3-23
- IBIB_jobdesc parameter, 3-9
- IBIB_jobdesc parameter for DSTACTIVE servlet, 3-36
- IBIB_jobdesc parameter for DSTLOG servlet, 3-41
- IBIB_jobdesc parameter for DSTRUNNOW servlet, 3-38
- IBIB_jobdesc parameter for DSTSCHED servlet, 3-24
- IBIB_jobname parameter for DSTSCHED servlet, 3-24
- IBIB_lastexec parameter for DSTLOG servlet, 3-41
- IBIB_location parameter for DSTDLMEM servlet, 3-18
- IBIB_mailcompany parameter for DSTSCHED servlet, 3-28
- IBIB_mailfrom parameter for DSTSCHED servlet, 3-29
- IBIB_mailhost parameter for DSTSCHED servlet, 3-29
- IBIB_mailsuffix parameter for DSTSCHED servlet, 3-29
- IBIB_message property, 9-2, 9-5
- IBIB_method parameter for DSTDLBULK servlet, 3-13
- IBIB_method parameter for DSTSCHED servlet, 3-24
- IBIB_name parameter for DSTDLBULK servlet, 3-13
- IBIB_name parameter for DSTDLLIST servlet, 3-20
- IBIB_name parameter for DSTDLMEM servlet, 3-18
- IBIB_notifyaddress parameter for DSTSCHED servlet, 3-31
- IBIB_notifybrief parameter for DSTSCHED servlet, 3-32
- IBIB_notifyflag parameter for DSTSCHED servlet, 3-25
- IBIB_notifyreply parameter for DSTSCHED servlet, 3-32
- IBIB_notifysubject parameter for DSTSCHED servlet, 3-32
- IBIB_parm parameter for DSTRUNNOW servlet, 3-38
- IBIB_parm parameter for DSTSCHED servlet, 3-25, 3-33
- IBIB_parname property, 9-5
- IBIB_position property, 9-5
- IBIB_postrpc1 parameter for DSTSCHED servlet, 3-25

-
- IBIB_postrpc2 parameter for DSTSCHED servlet, 3-25
 - IBIB_prerpc1 parameter for DSTSCHED servlet, 3-26
 - IBIB_prerpc2 parameter for DSTSCHED servlet, 3-26
 - IBIB_priority parameter for DSTRUNNOW servlet, 3-38
 - IBIB_recipients parameter for DSTDLBULK, 3-16
 - IBIB_recipients parameter for DSTDLBULK servlet, 3-13
 - IBIB_reply property, 9-2, 9-5
 - IBIB_repsub property, 9-5
 - IBIB_scheduleid parameter, 3-9
 - IBIB_scheduleid parameter for DSTACTIVE servlet, 3-36
 - IBIB_scheduleid parameter for DSTLOG servlet, 3-42
 - IBIB_scheduleid parameter for DSTRUNNOW servlet, 3-39
 - IBIB_sendformat parameter for DSTSCHED servlet, 3-26
 - IBIB_startdate parameter for DSTLOG servlet, 3-42
 - IBIB_startdate parameter for DSTSCHED servlet, 3-26
 - IBIB_starttime parameter for DSTLOG servlet, 3-42
 - IBIB_starttime parameter for DSTSCHED servlet, 3-27
 - IBIB_subject property, 9-2, 9-5
 - IBIB_tcpiplevel parameter for DSTACTIVE servlet, 3-36
 - IBIB_tcpiplevel parameter for DSTDLBULK servlet, 3-14
 - IBIB_tcpiplevel parameter for DSTDLMEM servlet, 3-18
 - IBIB_tcpiplevel parameter for DSTLOG servlet, 3-43
 - IBIB_tcpiplevel parameter for DSTRUNNOW servlet, 3-39
 - IBIB_tcpiplevel parameter for DSTSCHED servlet, 3-27
 - IBIB_toaddr property, 9-2, 9-5
 - IBIB_userid parameter, 3-7
 - IBIB_userid parameter for DSTDLBULK servlet, 3-14
 - IBIB_userid parameter for DSTDLMEM servlet, 3-19
 - IBIB_userid parameter for DSTRUNNOW servlet, 3-39
 - IBIB_value parameter for DSTDLMEM servlet, 3-19
 - IBIB_weekdays parameter for DSTSCHED servlet, 3-28
 - IBITRACE facility, A-2, A-3
 - identification arguments for subroutines, 4-7
 - identification parameters for ReportCaster Servlet API, 3-9
 - installation requirements for ReportCaster API, 1-2
 - INT function, 2-19
- ## J
- Java Database Connectivity (JDBC), 2-3, 3-3, 4-3
 - Java Development Kit (JDK), 1-2
 - JavaBeans components, 1-2, 2-1, 8-2, 8-4
 - JavaServer Pages (JSP) technology, 1-2, 2-21, 8-7 to 8-8
 - calling from HTML forms, 2-22 to 2-23
 - calling from URLs, 2-22
 - sample pages, 5-1 to 5-3, 9-1

JDBC (Java Database Connectivity), 2-3, 3-3, 4-3
JDK (Java Development Kit), 1-2
Job Description field, C-3 to C-4
Job Log, 10-2
 purging, 10-4
 viewing job status, 10-3
jobdesc argument for DSTRUN, 4-29
jobdesc argument for subroutines, 4-7
jobdesc_length argument for DSTRUN, 4-29
JSP (JavaServer Pages) technology, 1-2, 2-21, 8-7 to 8-8
 calling from HTML forms, 2-22 to 2-23
 calling from URLs, 2-22
 sample pages, 5-1 to 5-3, 5-6, 9-1

L

log files, 1-4, 3-40, 3-43, C-4
log reports, 3-40, 3-43, 5-16, 6-20
Log Summary reports, 5-16, 5-23
Log Summary window, 6-20
LOG_PURGE_PERIOD parameter, 10-4
logging on to ReportCaster API, 5-4, 6-3

M

mail parameters for DSTSCHED, 3-28
main.trc files, A-5
main.err tracing files, A-6
main.trc files, A-4
Maintain Distribution Lists file window, 6-15
Maintain Single Distribution Records window, 6-17
Managed Reporting Environment (MRE), 1-1, C-1
 accessing API-based server schedules, C-3
 Server Schedule component, C-3
 viewing an API-based log file, C-4

Managed Reporting Environment (MRE) users, C-1 to C-2
MAX_THREADS parameter, A-4
messages and codes, B-1
messages for ReportCaster Bean API, B-5
messages for ReportCaster Servlet API, B-5
messages for subroutines, B-2
method argument for DSTBULK, 4-12
method_length argument for DSTBULK, 4-12
methods, 1-2, 2-6 to 2-7, 2-12
monitoring Two-Way Email traffic, 10-5
MRE (Managed Reporting Environment), 1-1, C-1
 accessing API-based server schedules, C-3
 Server Schedule component, C-3
 viewing an API-based log file, C-4
MRE (Managed Reporting Environment) users, C-1 to C-2

N

name argument for DSTBULK, 4-10
name argument for DSTMEM, 4-20
name_length argument for DSTBULK, 4-10
name_length argument for DSTMEM, 4-20
No Contacts reports, 7-1, 7-8
 running, 7-2
non-bursted reports, 3-15 to 3-16
 creating an external file, 3-15 to 3-16
 generating text box input, 3-16
notification parameters for DSTSCHED, 3-31

O

owner argument for DSTMEM, 4-21
owner argument for DSTRUN, 4-30

P

packages, 1-2

parameters for ReportCaster Servlet API, 3-9

- DSTACTIVE, 3-35
- DSTDLBULK, 3-11
- DSTDLLIST, 3-19
- DSTDLMEM, 3-17
- DSTLOG, 3-40
- DSTRUNNOW, 3-38
- DSTSCHEM, 3-21, 3-28, 3-30 to 3-31

PARAM argument for DSTRUN, 4-30

parm_length argument for DSTRUN, 4-30

passing values to reports, 3-33 to 3-34

priority argument for DSTRUN, 4-29

procedure-based calls, 4-3, B-2

- DSTBULK, 4-8
- DSTMEM, 4-18
- DSTRUN, 4-25
- troubleshooting, B-2

Q

query strings, A-1, A-6 to A-7

R

rbabulk.htm form, 6-2, 6-14

rbabulkm.htm form, 6-2, 6-15

rbacopy.htm form, 6-2, 6-18

rbadlist.htm form, 6-2, 6-19

rbadlmem.htm form, 6-2, 6-17

rbalog.htm form, 6-2, 6-20

rbarunow.htm form, 6-2, 6-13

rbasched.htm form, 6-2, 6-6

rbastats.htm form, 6-2, 6-12

rcabook reports, 7-1, 7-7

- running, 7-2

- rcalert reports, 7-1, 7-9

 - running, 7-2

- rcaster_clist.jsp, 5-2, 5-23
- rcaster_copy.jsp, 5-2, 5-15
- rcaster_create.jsp, 5-2, 5-13
- rcaster_delete.jsp, 5-2, 5-16
- rcaster_detail.jsp, 5-2, 5-15
- rcaster_list.jsp, 5-2, 5-13
- rcaster_logdelete.jsp, 5-3, 5-24
- rcaster_loglist.jsp, 5-3, 5-13
- rcaster_logpdelete.jsp, 5-3, 5-24
- rcaster_logplist.jsp, 5-3, 5-23 to 5-24
- rcaster_newschedule.jsp, 5-2, 5-7, 5-13
- rcaster_property.jsp, 5-2, 5-14
- rcaster_run.jsp, 5-2, 5-16
- rcaster_runnowlog.jsp, 5-2, 5-24
- rcaster_runonce.jsp, 5-2, 5-13
- rcaster_setrun.jsp, 5-2, 5-16
- rcaster_update.jsp, 5-2, 5-15
- rcdlist reports, 7-1, 7-5

 - running, 7-2, 7-5

- rcnonote reports, 7-1, 7-8

 - running, 7-2

- rcnotify reports, 7-1, 7-7

 - running, 7-2

- rcuser reports, 7-1, 7-8

 - running, 7-2

- reader.err tracing file, A-6
- reader.trc files, A-4
- RECOVERY parameter, 10-5
- relative addresses, 3-8

 - specifying with DSTACTIVE servlet, 3-35
 - specifying with DSTDLLIST servlet, 3-19
 - specifying with DSTLOG servlet, 3-40

- relative addresses (*continued*)
 - specifying with DSTRUNNOW servlet, 3-37
 - specifying with DSTSCHED servlet, 3-21
 - ReportCaster Administrator user ID, 3-6, 3-35 to 3-36, C-1
 - ReportCaster Bean API, 1-1, 1-3, 2-1 to 2-2, 2-8
 - communication, 2-3
 - JavaServer Pages samples, 5-1
 - logging on, 5-4
 - messages, B-5
 - process flow, 2-8
 - security, 2-6
 - ReportCaster Bean API result sets, 2-18
 - number of rows, 2-19
 - retrieving column data, 2-18, 2-20
 - ReportCaster Log List DTD, 2-16
 - ReportCaster Log Process List DTD, 2-17
 - ReportCaster Process Log DTD, 2-17
 - ReportCaster Repository reports, 1-1, 7-1, 7-3
 - Address Book, 7-7
 - Alerts Schedules, 7-9
 - Distribution Type, 7-5
 - No Contacts, 7-8
 - ReportCaster Users, 7-8
 - running, 7-2
 - Schedule Contacts, 7-7
 - ReportCaster Repository tables, 2-3 to 2-5, 3-3 to 3-4, 4-2
 - ReportCaster Servlet API, 1-1, 1-4, 6-1 to 6-2, 6-5
 - calling from an HTML form, 3-8
 - DSTACTIVE, 3-2, 3-35
 - DSTDLBULK, 3-2, 3-10, 6-14
 - DSTDLLIST, 3-2, 3-19
 - DSTDLMEM, 3-2, 3-17
 - DSTLOG, 3-2, 3-40
 - DSTRUNNOW, 3-2, 3-37 to 3-38
 - DSTSCHED, 3-2, 3-20 to 3-21, 3-33
 - logging on, 6-3
 - messages, B-5
 - parameters, 3-9
 - security, 3-6
 - ReportCaster Users reports, 7-1, 7-8
 - running, 7-2
 - reports, 3-1, 3-20
 - running immediately, 1-4, 4-1 to 4-2, 4-25 to 4-26, 4-32, 6-5, 6-13
 - scheduling, 1-3, 3-1
 - scheduling with DSTSCHED, 3-20
 - setting status with DSTACTIVE, 3-35
 - requirements for ReportCaster, 1-2
 - result sets for ReportCaster Bean API, 2-18
 - number of rows, 2-19
 - retrieving column data, 2-18, 2-20
 - return codes, B-2
 - DSTBULK, B-5
 - DSTMEM, B-5
 - DSTRUN, B-3
 - translating into messages, B-2
 - returncode argument for DSTBULK, 4-15
 - returncode argument for DSTMEM, 4-23
 - returncode argument for DSTRUN, 4-31
 - Run a Scheduled Job Immediately window, 6-13
 - running ReportCaster Repository reports, 7-2
- ## S
- sample HTML forms, 3-9, 6-1 to 6-2, 6-5
 - sample JavaServer Pages, 5-1, 5-6, 9-1
 - log functions, 5-3
 - scheduling functions, 5-2
 - Schedule Contacts reports, 7-1, 7-7
 - running, 7-2
 - Schedule Detail DTD, 2-14
 - Schedule Directory, C-3
 - options, C-3 to C-4
 - properties, C-3 to C-4
 - Server Schedule component, C-3 to C-4
 - Schedule List DTD, 2-12
 - Schedule List window, 5-13
 - Schedule Properties DTD, 2-13

- scheduleid argument, 4-7
- scheduleid argument for DSTRUN, 4-28
- scheduleid_length argument for DSTRUN, 4-29
- scheduling reports, 1-3, 5-7
 - changing the status, 6-5, 6-12
 - creating schedules, 5-7
 - creating schedules using a JSP, 5-7
 - creating schedules using a servlet, 6-5 to 6-6
 - JavaServer Pages samples, 5-2
 - viewing schedules, 5-13
- SCHStop program, A-4
- security, 1-1
 - ReportCaster Bean API, 2-6
 - ReportCaster Servlet API, 3-6
 - subroutines, 4-3
- security for Two-Way Email, 8-7
- Self-Service Two-Way Email, 8-1, 9-4
 - Administrator console, 8-9, 10-1 to 10-2
 - creating Tool window, 9-5
 - interface, 8-9, 9-4
 - JavaBeans components, 8-2
 - JavaServer Pages samples, 9-1 to 9-2, 9-5
 - JSP processing, 2-21, 8-7 to 8-8
 - requirements, 8-2
 - security, 8-7
- Self-Service Two-Way Email Beans, 8-3
- Server Procedures component, C-3 to C-4
- Server Schedule component, C-3 to C-4
- servlet-enabled Web server, 1-2
- servlets, 1-2, A-2
 - accessing data sources, 3-3
 - calling from an HTML form, 3-8
 - DSTACTIVE, 3-2, 3-35
 - DSTDLBULK, 3-2, 3-10 to 3-11, 6-14
 - DSTDLLIST, 3-2, 3-19
 - DSTDLMEM, 3-2, 3-17
 - DSTLOG, 3-2
 - DSTRUNNOW, 3-2
 - DSTSCHED, 3-2
 - security, 3-6
 - tracing, A-2, A-4
- servlets (*continued*)
 - user IDs, 3-7
- SET command, 4-4, 4-6
- set methods, 2-12
- SET TRACEOFF=ALL setting, A-3
- SET TRACEON=ALL setting, A-3
- SET TRACEON=SNDH/1 setting, A-3
- SET TRACEON=SNDH/2 setting, A-3
- setAgentProperty method, 2-7
- setInitParameter method, 2-7
- setPass method, 2-6
- setting report status with DSTACTIVE, 3-35
- setUser method, 2-6
- specifying absolute addresses, 3-8
- specifying relative addresses, 3-8
- SQL-based tables, 2-3, 3-3 to 3-4, 4-2
- srv_userid argument for DSTBULK, 4-9
- srv_userid argument for DSTMEM, 4-18
- srv_userid argument for DSTRUN, 4-27
- srv_userid_length argument for DSTBULK, 4-9
- srv_userid_length argument for DSTMEM, 4-19
- srv_userid_length argument for DSTRUN, 4-27
- srv_userpass argument for DSTBULK, 4-9
- srv_userpass argument for DSTMEM, 4-19
- srv_userpass argument for DSTRUN, 4-27
- srv_userpass_length argument for DSTBULK, 4-9
- srv_userpass_length argument for DSTMEM, 4-19
- srv_userpass_length argument for DSTRUN, 4-27

- subroutines, 1-1 to 1-2, 1-4, 4-1, 4-3 to 4-4, A-2
 - accessing repository tables, 4-2
 - amper variables, 4-34
 - calling from a procedure, 4-3
 - calling using an HTML form, 4-7
 - calling using -SET, 4-4, 4-6
 - coding requirements, 4-6
 - DSTBULK, 4-2, 4-8, 4-18, 4-26
 - DSTMEM, 4-2, 4-18, 4-24
 - DSTRUN, 4-2, 4-25 to 4-26, 4-32
 - identification arguments, 4-7
 - messages, B-2
 - security, 4-3
 - tracing, A-2
 - user IDs, 3-7
 - using variables, 4-34

T

- tcplevel argument for DSTBULK, 4-15
- tcplevel argument for DSTMEM, 4-23
- tcplevel argument for DSTRUN, 4-31
- text box input, 3-16
- TRACEUSER parameter, A-2
- tracing error files, A-6
- tracing fex files, A-5
- tracing for ReportCaster API, A-1 to A-2, A-4
 - Distribution Server, A-4 to A-6
 - servlets, A-4
- translation procedures, B-2
 - DSTBLKER, B-5
 - DSTMEMER, B-5
 - DSTRUNER, B-3
- tw_focexec.jsp, 9-2

- tw_summary.jsp, 9-2
- tw_twoway.jsp, 9-2, 9-4 to 9-5
- Two-Way Email Administrator console, 8-1, 8-9, 10-1

U

- user argument for DSTBULK, 4-12
- user IDs, 1-3, 2-4, 2-6, 3-4, 3-6 to 3-7, 3-9, 3-14, 3-19 to 3-20, 3-31, 4-2 to 4-3, 5-4, 6-3
 - API-based, C-1 to C-2
- user_length argument for DSTBULK, 4-13
- user_length argument for DSTMEM, 4-21
- user_length argument for DSTRUN, 4-30

V

- validating users, 2-6, 3-6
- values, 3-33 to 3-34
- variables within a subroutine, 4-34
- viewing an API-based log file in MRE, C-4

W

- WebFOCUS cookie (WF_COOKIE), 1-3, 2-6, 3-6
- WebFOCUS logon, 1-1, 2-6, 3-6, 4-3
- WebFOCUS Reporting Server, 4-3, A-2
- WF_COOKIE (WebFOCUS cookie), 1-3, 2-6, 3-6

X

- XML result sets, 2-12

Reader Comments

In an ongoing effort to produce effective documentation, the Documentation Services staff at Information Builders welcomes any opinion you can offer regarding this manual.

Please use this form to relay suggestions for improving this publication or to alert us to corrections. Identify specific pages where applicable. You can contact us through the following methods:

Mail: Documentation Services – Customer Support
Information Builders, Inc.
Two Penn Plaza
New York, NY 10121-2898

Fax: (212) 967-0460

E-mail: books_info@ibi.com

Web form: <http://www.informationbuilders.com/bookstore/derf.html>

Name: _____

Company: _____

Address: _____

Telephone: _____ Date: _____

E-mail: _____

Comments:

Reader Comments