ibi™ WebFOCUS®

ReportCaster Legacy API Developer Reference

Version 9.3.3 | January 2025

Contents

Introducing the ReportCaster Java API

This section provides an overview of the ReportCaster Application Programming Interface (API) and describes the options available to customize your application.

About the ReportCaster API

The ReportCaster API provides access to ReportCaster functionality independent of ibi™ WebFOCUS® and ReportCaster user interfaces. Using the ReportCaster API, developers of self-service applications can enable users to distribute reports using web applications.

Using JPA, the ReportCaster API accesses the WebFOCUS® repository to read and write scheduling and distribution information. The API bundles the distribution information and sends it to the ibi™ WebFOCUS® Reporting Server to run. Developers must provide a valid ReportCaster sign-in ID to their ReportCaster API applications for ownership of schedules and Distribution Lists.

In addition to the functionality available through the ReportCaster API, ReportCaster retains the functionally stabilized ReportCaster Servlet API for Address Book functionality and ReportCaster API subroutines, which are C-based components of the WebFOCUS® Reporting Server. ReportCaster also retains the functionally stabilized Bean API for use only with legacy applications.

As of Version 4 Release 3.5, the ReportCaster Servlet API is functionally stabilized. For Version 5.3 and higher, including Version 7 Release 7, Information Builders recommends that you use the ReportCaster API to implement all ReportCaster functionality. In Release 8.0 Version 05, Information Builders recommends the use of RESTful web services to implement ReportCaster functionality.

How This Manual Works

This manual provides information to supplement that found in the ReportCaster API Java document. It includes a description of the packages available in the API and an outline of the samples provided with the API, which are provided as a tool for understanding how to use the API. In addition, this manual provides information on Servlet API information for z/OS users and legacy Subroutine API information is provided as a reference for users who have applications developed in earlier releases.

For information on the ReportCaster API Java documentation, contact Customer Support.

Installation Requirements

The following are installation requirements for the ReportCaster API:

- Servlet-enabled web server or application server.
- JDK 1.6 on your web server or application server.

For more information on ReportCaster requirements, see the *ibi™ WebFOCUS® Installation and Configuration* manual for your platform.

ReportCaster Overview

ReportCaster is a scheduling and distribution application that centralizes the execution and distribution of WebFOCUS reports, the contents of URLs, and files. ReportCaster supports multiple administrators and provides a single point of control for managing the information required to run an organization.

ReportCaster Components

The following aspects of ReportCaster can be configured using the ReportCaster API:

ReportCaster Address Book. An Address Book is a list of addresses to which ReportCaster output is sent. Configuration of Address Book components is managed through the ibi.broker.api.data.addrbook package.

ReportCaster Scheduling. Scheduling is a means by which report output is distributed according to a specified time table. Reports can be distributed through email, a printer, a report in a Managed Reporting folder, or the Report Library.

Report Library. The Report Library enables you to store and manage content distributed by ReportCaster for rapid retrieval and future use.

ReportCaster Console. The ReportCaster Console is an interface made up of a set of HTML pages and JavaServer Pages (JSPs) that enable ReportCaster Administrators to manage ReportCaster processing.

ReportCaster API Components

The ReportCaster API is comprised of the following parts:

ReportCaster API. This is the most current, comprehensive ReportCaster API, which offers the full range of ReportCaster functionality including scheduling, log, Address Book, and Report Library.

The ReportCaster API is fully upward compatible with the ReportCaster Servlet, and Subroutine APIs, and is partially backward compatible.

Functionally Stabilized ReportCaster Servlet API. The ReportCaster Servlet API.

For more information, see ReportCaster Servlet API.

Functionally Stabilized ReportCaster Subroutine API. The ReportCaster subroutine API.

For more information, see ReportCaster API Subroutines.

ReportCaster API Functionality

The ReportCaster API consists of packaged Java classes published by Information Builders. The ReportCaster API enables developers to use JavaServer Pages (JSP) technology or a Java application to create an independent, customized application that offers the features of ReportCaster.

Address Book Options

The ReportCaster API provides methods that enable you to:

- Create a new Address Book.
- Add a line to an existing Address Book, which includes the destination address and burst value.

- Retrieve an Address Book from a repository.
- Retrieve a list of all public Address Books for a specific distribution method.
- Retrieve a list of all private Address Books belonging to a specific owner for a specific distribution method.
- Update an existing Address Book by adding a new destination or burst value.
- Delete an Address Book.
- Delete a line from an Address Book, which includes the destination address and burst value.

Scheduling Options

The ReportCaster API provides methods that enable you to:

- Create a report schedule specifying the run time, distribution method (email, FTP, printer, Managed Reporting, or Report Library), notification parameters, report format, and whether to burst the report.
- Schedule alerts.
- Schedule the distribution of a file or the contents of a URL.
- Schedule a Managed Reporting procedure.
- Retrieve a list of scheduled Managed Reporting procedures.
- List the procedures (FOCEXECs) on the default WebFOCUS Reporting Server available for scheduling.
- Specify whether to use a Distribution List, distribution file name, single destination, or dynamic address list.
- Specify whether the report output should be sent within the body of the email (inline) or as an attachment (email distribution only).
- Update a schedule.
- Delete a schedule.
- Retrieve a list of existing schedules.
- Retrieve the properties and details of a schedule.

- Run a job once at a specific time without storing the schedule information in the WebFOCUS repository.

Report Library Options

The ReportCaster API provides methods that enable you to:

- Distribute reports to the Report Library.
- Manage content of the Report Library.
- Track versioning.
- View content in the Report Library.
- Create an Access List.
- Manage the contents of an Access List.

ReportCaster Console Options

The ReportCaster API provides methods that enable you to:

- Query the ReportCaster Distribution Server queue to show a list of pending jobs.
- Query the ReportCaster Distribution Server queue to show a list of running jobs.
- Change priority of a job from the default (3).
- Delete a pending job from the queue.
- Suspend and Resume the Distribution Server.
- Get the status of the Distribution Server.

Log File Options

The ReportCaster API provides methods that enable you to:

- Retrieve the list of schedules that have log information stored in the WebFOCUS repository.

- Retrieve log information for a specific schedule.
- Retrieve log information based on log field values and specified conditions.
- Purge log records for a specified schedule.
- Purge log records based on log field values and specified conditions.

ReportCaster API Authentication and Ownership

User IDs that access ReportCaster must exist in the WebFOCUS repository prior to accessing the ReportCaster API components. ReportCaster Administrators may migrate user IDs from a prior release (see the *ibi™ WebFOCUS® Installation and Configuration* manual for your platform), or create new user IDs.

Command line API programs authenticate using the IBI_Authentication_Type setting in the WebFOCUS Administration Console. For example, if this setting is Internal, then you must supply a valid user and password. These can be found in the WebFOCUS repository. For more information about this setting, see the *ibi™ WebFOCUS® Security and Administration* manual.

ReportCaster uses the user ID for ownership of schedules and Distribution Lists created using the ReportCaster API components.

There are two types of ReportCaster API users, Administrator and Users.

Create users and groups to provide ReportCaster capabilities for non-Managed Reporting users (when Managed Reporting is configured with ReportCaster, you must use the Managed Reporting User Administration tool to create and manage users and groups).

Manage environment configuration settings using the ReportCaster Server Configuration tool.

If you have Administrator privileges, you can:

- Create and maintain public or private Distribution Lists owned by any user.
- Maintain, run, and view schedules owned by any user.
- Create your own schedule.

- Access the Library Access List within the Report Library product.
- Track the status of a schedule.
- View and delete schedule log information.

If you have User privileges and have been assigned scheduling capabilities, you can:

- Create and maintain your own public or private Distribution Lists.
- View and use public Distribution Lists owned by another user.
- Create, maintain, or view your own schedules.
- Run your own schedules.

ReportCaster API Packages

ReportCaster API packages classes that provide the following functionality:

- **ibi.broker.api** provides the primary means for obtaining connections to ReportCaster.
- **ibi.broker.api.cci** provides high-level management via the Address Book Manager, the Console Manager, Library Content Manager, Schedule Manager, and User Manager.
- **ibi.broker.api.data** allows authentication with Web Services applications.
- **ibi.broker.api.data.addrbook** allows you to create Address Books and specify Destination Lists.
- **ibi.broker.api.data.console** allows you to manage the particulars of individual jobs.
- **ibi.broker.api.data.dslog** allows the encapsulation of log information.
- **ibi.broker.api.data.library** allows you to set specifications for the ReportCaster Library.
- **ibi.broker.api.data.libraryaccess** allows you to set specifications for Library Access Lists.
- **ibi.broker.api.data.schedule** allows you to create and modify schedules.
- **ibi.broker.scheduler.plugin** provides the ability to implement plug-in options.
- **ibi.broker.api.ws** provides Web Services specific API calls.

ReportCaster API Development at a Glance

In order to create API code that employs ReportCaster functionality, the API calls have to be executed in a specified order.

To create API code, complete the following steps:

- Create a connection object, which allows you to indicate crucial server connection information.
- Create a manager object, which allows you to specify at a high level which aspect of ReportCaster you will be affecting. For example, implement ScheduleManager to create a new schedule or access an existing schedule.
- Implement the classes that are associated with the manager object you have selected. At the class level, you can inspect and set the properties of individual instances of ReportCaster elements. For example, the destination class in the schedule package allows you to determine the destination list of an existing schedule.

Creating a Connection Object

Creating a connection object is the first step to writing API code. The connection object contains authentication information and context information which includes the repository location and ReportCaster Distribution Server settings.

In most cases, the code for creating a connection object is repeated, meaning that this code can be written into a convenience class or re-used. The only variables that need to be changed are host, port, user and password. Additionally, coders have a choice of authentication types: PasswordCredential, WebAuthenticateCredential, and SecurityTokenCredential. WebAuthenticateCredential allows authentication through the Web Server and SecurityTokenCredential allows authentication through cookies.

The code for creating a connection object conforms to the Sun CCI – Common Connector Interface. This architecture gives Information Builders and users the eventual flexibility to extend the product to integrate with other applications that also conform to this standard, as well as eventually integrate with features in J2EE application servers.

Create a Connection Object

The following code demonstrates how to create a connection object using password authentication via PasswordCredential.

```
CasterManagedConnectionFactory managedConnectionFactory =
  new CasterManagedConnectionFactory();
managedConnectionFactory.setServerName(host);
managedConnectionFactory.setPortNumber(port);
CasterConnectionFactory connectionFactory =
  (CasterConnectionFactory)managedConnectionFactory.
  createConnectionFactory();
ConnectionSpec credential =
  new PasswordCredential(user, password.toCharArray());
CasterConnection myConnection =
  ((CasterConnection)connectionFactory.getConnection(credential));
out.println("connected");
```

Creating a Manager Object

Once the connection object is created, you need to create at least one manager object. The ReportCaster API manager objects provide six areas of functionality: scheduling, Address Book management, user management, log management, Report Console use and library administration. Each of these areas of functionality has a manager that creates and controls the functionality, and a related package that contains the classes that handle the details of the functionality.

For example, you would use the ScheduleManager object to return a list of schedules, but you would use the classes in the ibi.broker.api.data.schedule package to manipulate the details of an individual schedule. Additionally, multiple manager objects can be created from the same connection. For example, an AddressBookManager can co-exist with a ScheduleManager.

Creating a ScheduleManager Object

The following syntax demonstrates how to create a ScheduleManager Object.

```
//Create ScheduleManager
ScheduleManager manager = myConnection.getScheduleManager();
```

Using Methods

Once a manager object is created, the methods that belong to that manager can be used to access or create the functionality that is required.

Use Schedule Manager Methods

The code below demonstrates how to create a Schedule object. This is one of the detail classes in the ibi.broker.api.data.schedule package that creates a schedule with default settings, such as RUN ONCE and Priority 3, as well as whatever defaults are listed in dserver.xmls.

The other functionality required in a schedule, such as Tasks and Distribution settings, can be added to the schedule object, as well.

```
Schedule schedule = manager.createScheduleInstanceDefault();
//set description
schedule.setDescription(scheduleDescription);
//set distribution to library
StorageLibrary distribution = new StorageLibrary();
schedule.setDistribution(distribution);
//create task
TaskWFServerProcedure task = new TaskWFServerProcedure();
task.setProcedureName(fexFileName);
task.setExecId(execId);
task.setExecPassword(execPassword);
schedule.setTaskList(new Task[]{task});
```

Once the components of the schedule are set, it needs to be written to the database.

```
//subscribe schedule
manager.addSchedule(schedule);
```

Customizing Error Messages

You can manually customize ReportCaster API error messages by editing the messages in the properties files listed in the table below.

Procedure

1. Copy the properties files you need from `ReportCaster93\lib\webfocus-intl-9.3.0.jar`. These files are located in the `ibi/intl/` directory in the `ReportCaster93\lib\webfocus-intl-9.3.0.jar`.
2. Manually edit the messages in the new bundle file.
3. Ensure to put the new jar file path before of `webfocus-caster-server-9.3.0.jar` in the classpath.

File Name	Contains Error Messages For...
CasterMessages.properties	Entire API
ScheduleMessages.properties	Schedules
AddressBookMessages.properties	Address Books
ExecIdMessages.properties	Execution IDs
LibraryAccessMessages.properties	Library Access Lists
LibraryContentMessages.properties	Report Library
LogMessages.properties	Schedule logs
UserMessages.properties	User Management

Within each properties file, each error message has an associated error code. The error messages are written in English, by default, but can be customized in another language. If you want to use a non-English language, you must modify the appropriate properties file. For example, for French, you would modify `CasterMessages_fr.properties`.

The steps for changing the properties files depend on whether you are redeploying the open directory structure, or repackaging and redeploying the web archive (.war) file. For more information about the steps involved for each scenario, see the *ibi™ WebFOCUS® Security and Administration* manual.

Additional Information

This section contains additional information in support of the API.

WebFOCUS Components Used by the Legacy ReportCaster Java API

When the sample API programs are compiled and run from the ibi\WebFOCUS8x\ReportCaster\Samples\Tools directory, all of the components of WebFOCUS are available to the API. If you are using the API from another location, you must ensure that the following components are available to the API:

- ...ibi\WEBFOCUS93\config
- ...ibi\WEBFOCUS93\client
- ...ibi\WEBFOCUS93\utilities
- ...ibi\WEBFOCUS93\ibi_html (needed if StyleSheets under this location are included in the scheduled procedure)
- ...ibi\WEBFOCUS93\ReportCaster\lib

Set the variable IBI_DOCUMENT_ROOT to the path to ibi\WebFOCUS9x in your remote location. The classpath must include webfocus-caster-server-9.3.0.jar located in the ...ibi\WebFOCUS\ReportCaster93\lib. You must also include the JDBC driver jar file name in the classpath. The other jar files from ...ibi\WebFOCUS\ReportCaster93\lib are in the Java Manifest Class Path in webfocus-caster-server-9.3.0.jar.

ReportCaster Job Log Error Codes

- **0** = Success
- **1** = Error

- **2** = Warning
- **6** = The Job is executing and no errors have occurred.
- **7** = The Job is executing and an error has occurred.

Note: In earlier releases, error codes 6 and 7 do not occur because Log information is not written to the Log tables until a job has finished. In Release 8, Log information is written to the Log tables during Job execution so the new codes have been added to distinguish running jobs from jobs that have completed.

ReportCaster Content in the WebFOCUS Repository

In Release 8, the ReportCaster Java API is intended for migrated applications. When applications are migrated to Release 8, ReportCaster content is moved to the WebFOCUS Repository in the following manner. Schedules, Distribution Lists, and Access Lists are placed in My Content folders under the top-level folder named ReportCaster. Library reports that are the output of scheduled Managed Reporting procedures are placed in a Library Content folder under the domain where the migrated Managed Reporting procedure is located. Library reports that are the output of scheduled WebFOCUS Reporting Server procedures are placed in Category folders under a top-level Library Content folder. The legacy API does not require that you specify a specific folder when content is created, so by default, when new Schedules, Distribution Lists, Access Lists, and Library Reports are created using the ReportCaster API, the above locations are the default locations for this content.

The default folders for legacy API output are created during the migration process. If migration has not taken place, you must create these folders manually prior to using the ReportCaster API or specify alternate locations for the content. For more information, contact Customer Support.

ReportCaster Java API Samples

Information Builders offers sample API programs as a learning tool, which are not intended for production use. Licensed users are welcome to alter and extend these samples and deploy them in other environments as they see fit. Information Builders will support the documented functionality of its API classes and methods. However, Information Builders is not responsible for the functionality or behavior of products built with its API unless the documented or expected behavior of its discrete classes and methods is different than the actual results.

While these samples can be implemented, compiled, and run in numerous ways, batch files are included that help compile and run these samples.

Introduction to API Samples

These topics describe the ReportCaster API sample applications. They contain 46 Java applications and two Java Server Pages (JSP) that illustrate the usage of the API. Each application or JSP offers a discrete example of a piece of functionality that is available in the API. These samples are available in *drive:*\ibi\WebFOCUS82\ReportCaster\samples.

Using Batch Files to Compile Samples

Information Builders encourages developers to alter and recompile these applications, as needed. However, in order to make this more convenient, batch files are included with the product to assist in the compilation and execution of these samples. The batch files were written and tested in a Windows environment.

In order to run the batch files in UNIX, change the extension of the executable from .bat to the accepted scripting extension in your environment (for example, .sh) and change the call command in environment.bat to the proper command in your environment.

Note: The batch files are documented for a Windows environment, but are also provided for a UNIX environment.

Compiling Sample Files

SamplesComp.bat is provided to automatically compile all of the .java files included with the samples. It automatically places the generated class files in a directory called CompiledClasses. Developers can feel free to modify or enhance this script to place the compiled classes in any directory indicated.

Before using SamplesComp.bat, you must first modify the settings that are specific to your environment.

Configure the SamplesComp.bat Batch File

This section defines the contents and variable descriptions for the SamplesComp.bat file. An example is also provided.

```
set JAVA_RUN=[jvmpath]
set CLASSPATH= .; [jarpath]; )
%JAVA_RUN% -g [root]/ibi/WebFOCUS93/ReportCaster/Samples/*.java -d
[root]/ibi/WebFOCUS93/ReportCaster/Samples/Tools/CompiledCode
```

where:

jvmpath

Is the path to the javac executable (that is javac.exe) associated with your JVM. For example:

```
C:\Program Files\Java\jdk1.6.0_17\bin\javac
```

jarpath

Is the path to the webfocus-caster-server-9.3.0.jar files. For example:

```
C:\ibi\WebFOCUS\ReportCaster93\lib\webfocus-caster-server-9.3.0.jar;
```

Note: The *jarpath*, which is included with the API samples and used in the examples, points to webfocus-caster-server.jar in the /lib directory of the Distribution Server. This directory contains all of the other necessary jar files to enable the ReportCaster API to work with WebFOCUS. You may need to modify the classpath to include other required .jar files. If your API application does not have access to these directories, you must include webfocus-caster-server.jar and the other jar files in your application.

root

Is the directory structure prior to the /ibi directory on your machine. You need to replace this value twice in the string after the JAVA_RUN command is invoked.

For example:

```
set JAVA_RUN="C:\Program Files\Java\jdk1.6.0_17\bin\javac"
set CLASSPATH= .;C: \ibi\WebFOCUS93\ReportCaster\lib\webfocus-caster-
server-9.3.0.jar
%JAVA_RUN%  -g C: /ibi/WebFOCUS93/ReportCaster/Samples/*.java -d C:
/ibi/WebFOCUS93/ReportCaster/Samples/Tools/CompiledCode
```

Running the Samples

There are multiple sample applications provided for various functions. Five batch files are provided to assist in running these applications. The batch files are named according to function and include:

- Address Book (AddrBook.bat)
- Schedule (Schedule.bat)
- Console (Console.bat)
- Log (Log.bat)
- Library Content (LibraryContent.bat)

Each of these batch files reference a batch file named setenv.bat, which will set environment specific settings, such as the location of the Java executable and the classpath. In order to run these batch files, this will need to be configured in a similar fashion as SamplesComp.bat.

Configure the setenv.bat Batch File

This section provides the contents of and variable descriptions for the setenv.bat file.

Note:

- In order for the setenv.bat file to work, you must uncomment the lines that contain the following variables: SET IBI_DOCUMENT_ROOT, SET APPL_PATH, and SET JAVA_HOME.
- This example shows you how to run the API from the machine on which WebFOCUS is installed. If you are running the API on a machine where WebFOCUS is not installed, you must copy the files to the new location. For more information, contact Customer Support.

```
@echo off
REM Copyright 1996-2014 Information Builders, Inc. All rights reserved
REM -- uncomment the line below and replace with the install root of
WebFOCUS
rem set IBI_DOCUMENT_ROOT=c:\ibi\WebFOCUS82
REM -- optional: change the line below and replace with your application
classpath
rem set APPL_PATH=.\CompiledCode
REM -- optional: replace default WebFOCUS Java Home
rem set JAVA_HOME=C:\Program Files\Java\jre6
REM -- optional: used to echo back environment variables
rem set ECHO=ON
```

where:

IBI_DOCUMENT_ROOT

Is the root location of the WebFOCUS Client installation. For example, C:\ibi\WebFOCUS82.

APPL_PATH

Is the location of the compiled API samples programs or class files.

JAVA_HOME

Is the location of the Java installation. For example, C:\Program Files\Java\jre6.

> **Note:** When the migrate utility is run upon upgrading from releases prior to WebFOCUS 8, folders are created to store the schedules. When a new WebFOCUS 8 version is installed without migrating, you can use the following procedure to ensure that the requisite folders are created.

Create Folders Required for the API Samples

Procedure

1. In the WebFOCUS BI Portal, right-click **Content** in the Resources tree.
2. Click **New** and then click **Folder**.
3. In the Title field, enter *ReportCaster* and click **OK**.
4. Right-click the ReportCaster folder and click **Properties**.
5. Select the **Auto Create My Contents Folder** check box and click **OK**.
6. Right-click the ReportCaster folder again and click **Publish**.

   > **Note:**
   > If this task is not performed, then certain sample API programs like S06_Add_Notification will display the following error message:
   >
   > ```
   > Failed to find the folder IBFS:/.../~admin
   > ```

7. In the WebFOCUS BI Portal, right-click **Content** in the Resources Tree.
8. Click **New** and then click **Folder**.
9. In the Title field, enter *Library Content* and click **OK**.
10. Right-click on the Library Content folder and click **Publish**.

Additional Files Packaged With the Samples Application

The following additional files are available with the ReportCaster API samples package.

samples.properties

Instead of having to change the authentication information contained in Util.java and then recompile, samples.properties contains the user name, user password, ReportCaster Distribution Server host name, and ReportCaster Distribution Server port. If this file is not found at run time, the applications will use the default authentication values coded in Util.java.

Note: In Release 8.0 Version 01 and higher, it is no longer necessary to supply the Distribution Server host and port.

Running the Legacy Java API on a Machine Without a WebFOCUS Installation

Note: This process requires that you have access to a WebFOCUS installation.

You can run the Legacy Java API on a machine where WebFOCUS is not installed. In order to do this, you must copy configuration files from a machine where WebFOCUS is installed, and then make changes as defined in the procedure below. You can perform this process using a Windows installation.

In WebFOCUS 8, the ReportCaster Legacy Java API calls the IBFS security system to authenticate and authorize users, and to secure resources accessed during schedule creation and execution. Therefore, any files needed by the IBFS security system must be available to an application calling the ReportCaster Legacy Java API. The sample scripts provided in the ibi\WebFOCUS82\ReportCaster\samples\tools directory are set up so that the sample API programs can be:

- Compiled from this directory by modifying the SamplesComp script.
- Run from this directory by modifying the setenv script.

Note: The Legacy Java API is functionally stabilized and is intended for migrated applications. Restful Web Services is recommended for new development.

To configure a machine to run the Legacy Java API without WebFOCUS installed:

1. Create an ibi\WebFOCUS82 folder in the new location.
2. Copy the following folders from the WebFOCUS installation and paste them into the new location:
   - ibi\WebFOCUS82\client
   - ibi\WebFOCUS82\config
   - ibi\WebFOCUS82\utilities\bin
   - ibi\WebFOCUS82\utilities\setenv
   - ibi\WebFOCUS82\utilities\logger
   - ibi\WebFOCUS82\ReportCaster
3. Edit the ibi\WebFOCUS82\config\webconfig.xml in the new location and globally change the path to ibi\WebFOCUS82 to match the new location.
4. Edit the ibi\WebFOCUS82\utilities\setenv\utilprodvars.bat in the new location and do the following:
   a. Change WFROOT to the new location.
   b. Change WEBinf to %WFROOT%\ReportCaster
5. Edit the ibi\WebFOCUS82\utilities\setenv\utiluservars.bat in the new location and set the path to Java and to the JDBC driver.
6. Compile and run the sample Legacy Java API programs as described in this manual.

Address Book Samples

The Address Book sample folder contains Java applications that show how to create and manipulate ReportCaster Address Books using the API. As with all of the samples, it uses the ReportCaster authentication and context information contained in the **samples.properties** directory. These sample Java applications are presented as examples for learning purposes and are not represented as production quality applications.

A01_Add_Book.java

This application creates a new Address Book by accepting five parameters from the command line: owner, book name, book access type (for example, PUBLIC, PRIVATE), distribution method (for example, FTP, EMAIL, PRINT), and location (which is the destination of the first entry of the Address Book). The code can be modified to also require burst value.

Usage: java A01_Add_Book <owner> <bookName> <bookAccess> <bookMethod> <location>

A02_Add_Element.java

This application adds a line to an existing Address Book.

It accepts three arguments: The name of the Address Book, the burst value, and the distribution location.

The usage is as follows:

Usage: java A02_Add_Element <bookName> <burstValue> <location>

A10_Get_Book.java

This application retrieves an Address Book from the repository and writes the contents to standard out. It accepts one argument: the name of the book that is to be retrieved.

Usage: java A10_Get_Book <bookName>

A11_Get_PublicList.java

This application returns a list of all public Address Books for a specific distribution method, and their owners. It accepts one argument: the method of distribution (for example, FTP, EMAIL, PRINT).

Usage: java A11_Get_PublicList <bookMethod>

A12_Get_PrivateList.java

This application returns a list of all private Address Books belonging to a specific owner for a specific distribution method. It accepts two arguments: the owner and the method of distribution (for example, FTP, EMAIL, PRINT).

Usage: java A12_Get_PrivateList <owner> <bookMethod>

A13_Get_OwnerList.java

Writes a list of Address Book owners to standard out.

Usage: java A13_Get_OwnerList

A31_Update.java

This application updates an existing Address Book by adding a new location. It accepts two arguments: the name of the book to be updated and the location that is to be added. The code can be modified to also accept burst values for the update.

Usage: java A31_Update <bookName> <location>

A32_Delete_Book.java

This application deletes an Address Book. It accepts one argument: the name of the Address Book to be deleted.

Usage: java A32_Delete_Book <bookName>

A33_Delete_Element.java

This application deletes a line from an Address Book. It accepts three arguments: the name of the Address Book, the burst value from the line to be deleted, and the location (destination) from the line to be deleted.

Usage: java A33_Delete_Element <bookName> <burstValue> <location>

Library Access List Samples

The Library Access List sample folder contains Java applications that show how to create and manipulate ReportCaster Access Lists using the API. As with all of the samples, it uses the ReportCaster authentication and context information contained in the **samples.properties** directory. These sample Java applications are presented as examples for learning purposes and are not represented as production quality applications.

AL_Add_AccessList.java

This application adds a new access list containing a single user to the WebFOCUS repository. It accepts four arguments: owner, access list name, access list description, and the user on the list.

Usage: java AL_Add_AccessList <owner> <name> <description> <user>

AL_Add_AccessElement.java

This application adds a user to an existing access list in the WebFOCUS repository. It accepts four arguments: owner, access list name, access list description, and the user on the list.

Usage: java AL_Add_AccessElement <owner> <name> <description> <user>

AL_Update_AccessList.java

This application replaces an existing user and burst value in an existing access list. It accepts three arguments: the access list name, the new user, and the new burst value.

Usage: AL_Update_AccessList <name> <user> <burst value>

AL_Update_AccessElement.java

This application adds a user and burst value to an existing access list. It accepts three arguments: the access list name, the new user, and the new burst value.

Usage: AL_Update_AccessElement <name> <user> <burst value>

AL_Get_AccessList.java

This application prints the name, description, and owner of an existing access list. It accepts one argument: the access list name.

Usage: java AL_Get_AccessList <name>

AL_Get_AccessElement.java

This application prints the name, description and owner of an existing access list and lists the user and burst values in the access list. It accepts one argument: the access list name.

Usage: java AL_Get_AccessElement <name>

AL_Delete_AccessList.java

This application deletes an existing access list from the WebFOCUS repository. It accepts one argument: the access list name.

Usage: java AL_Delete_AccessList <name>

AL_Delete_AccessElement.java

This application deletes a member from an existing access list in the WebFOCUS repository. It accepts two arguments: the access list name and the member to be deleted.

Usage: java AL_Delete_AccessElement <name> <member>

Schedule Samples

This section provides sample Java applications that show how to create and manipulate ReportCaster schedules using the API. As with all of the samples, it uses the ReportCaster authentication and context information contained in samples.properties. These sample Java applications are presented as examples for learning purposes and are not provided as production quality applications.

S01_Add_Library.java

This sample creates a default schedule instance (a run once, email, WebFOCUS Reporting Server procedure) and modifies it to distribute to the library. It accepts four arguments: the Job Description, the name of the FEX on the WebFOCUS Reporting Server, the execution ID for access to the WebFOCUS Reporting Server, and the password for access to the WebFOCUS Reporting Server.

Usage: java S01_Add_Library <scheduleDescription> <fexFileName> <execId> <execPassword>

S02_Add_Email.java

This sample creates a default schedule instance of a WebFOCUS Procedure through email. It accepts five arguments: the Job Description, the name of the FEX on the WebFOCUS Reporting Server, the execution ID for access to the WebFOCUS Reporting Server, the password for access to the WebFOCUS Reporting Server, and a single email address for distribution.

Usage: java S02_Add_Email <scheduleDescription> <fexFileName> <execId> <execPassword> <emailAddress>

S02_Add_FTP.java

This sample creates a default schedule instance of a WebFOCUS Procedure distributed through FTP. It accepts ten arguments: the Job Description, the name of the FEX on the WebFOCUS Reporting Server, the execution ID for access to the WebFOCUS Reporting Server, the password for access to the WebFOCUS Reporting Server, a single FTP destination address, the FTP location, the FTP server name, the FTP user id, the FTP password, and a flag to set Secure FTP to Y or N.

Usage: java S02_Add_FTP <scheduleDescription> <fexFileName> <execId> <execPassword> <singleAddress> <sftpLocation> <sftpServerName> <sftpUser> <sftpPassword> <sftpEnabled (Y,N)>

S03_Add_StandardReport.java

This sample creates a default schedule and modifies it to distribute a Managed Reporting Standard Report to the library. It also queries the FEX in Managed Reporting for parameter values and shows how to submit these values into the schedule. It accepts five arguments: the Job Description, the Folder HREF in Managed Reporting that contains the FEX, the HREF representing the name of the FEX that is stored in Managed Reporting, the execution ID for access to the WebFOCUS Reporting Server, and the password for access to the WebFOCUS Reporting Server.

Usage: java S03_Add_StandardReport <scheduleDescription> <folderHREF> <procedureHREF> <execId> <execPassword>

S04_Add_AmperParameter.java

This sample creates a schedule that distributes a Managed Reporting Standard Report to the library. It queries the FEX in Managed Reporting for parameter values and then shows how to use the optionValue class to obtain an acceptable range of input values for the parameters. It accepts five arguments: the Job Description, the Folder HREF in Managed Reporting that contains the FEX, the HREF representing the name of the FEX that is stored in Managed Reporting, the execution ID for access to the WebFOCUS Reporting Server, and the password for access to the WebFOCUS Reporting Server.

Usage: java S04_Add_AmperParameter <scheduleDescription> <folderHREF> <procedureHREF> <execId> <execPassword>

S05_Add_TimeInfo.java

This sample creates a default schedule (a run once, email, WebFOCUS Reporting Server procedure) and changes it to distribute library and run every minute. It accepts four arguments: the Job Description, the name of the FEX on the WebFOCUS Reporting Server, the execution ID for access to the WebFOCUS Reporting Server, and the password for access to the WebFOCUS Reporting Server.

Usage: java S05_Add_TimeInfo <scheduleDescription> <fexFileName> <execId> <execPassword>

S06_Add_Notification.java

This sample creates a default schedule (a run once, email, WebFOCUS Reporter Server procedure) and changes it to distribute library. It then turns on full and brief notification for all distributions. It accepts five arguments: the Job Description, the name of the FEX on the WebFOCUS Reporting Server, the execution ID for access to the WebFOCUS Reporting Server, the password for access to the WebFOCUS Reporting Server, and the email address that will be used for both the full and brief notification.

Usage: java S06_Add_Notification <scheduleDescription> <fexFileName> <execId> <execPassword> <notificationAddress>

S07_Add_Alert.java

This sample creates a default schedule and modifies it to distribute a Managed Reporting Standard Report to the library, run in minute intervals, and schedule it as an alert. It accepts five arguments: the Job Description, the Folder HREF in Managed Reporting that contains the FEX, the HREF representing the name of the FEX that is stored in MR, the execution ID for access to the WebFOCUS Reporting Server, and the password for access to the WebFOCUS Reporting Server.

Usage: java S07_Add_Alert <scheduleDescription> <folderHREF> <procedureHREF> <execId> <execPassword>

S10_Get_ScheduleList.java

This sample displays a list of schedules owned by the designated owner to standard out. It accepts one argument: the owner of the schedules.

Usage: java S10_Get_ScheduleList <owner>

S11_Get_OwnerList.java

This sample returns a list of all of the owners of schedules in the repository to standard out.

Usage: java S11_Get_OwnerList

S21_Run.java

This sample runs a specified schedule. It accepts two arguments: the owner of the schedule and the Job Description.

Usage: java S21_Run <owner> <scheduleDescription>

S31_Delete.java

This sample deletes a specified schedule. It accepts two arguments: the owner of the schedule and the Job Description.

Usage: java S31_Delete <owner> <scheduleDescription>

S31_Update.java

This sample accepts an owner and a Job Description and updates the frequency information in the specified schedule. This application can be modified to accept multiple parameters for update, or to update different fields or values. It accepts two arguments: the owner of the schedule and Job Description.

Usage: java S31_Update <owner> <scheduleDescription>

Library Samples

LibraryViewerSample.java

This application gets the categories, descriptions, and versions in the Report Library and displays the list. It accepts no arguments.

Usage: java LibraryViewerSample

L04_Get_AccessibleVersion.java

This application gets the first version of a report in the Report Library and writes it to a file. It accepts three arguments: the user, the report description, and the output filename.

Usage: java L04_Get_AccessibleVersion <user> <report description> <filename>

Console Samples

The Console directory contains sample Java applications that show how to use the ReportCaster administrator console for the API. As with all of the samples, it uses the ReportCaster authentication and context information contained in samples.properties. These sample Java applications are presented as examples for learning purposes and are not represented as production quality applications. Note that with the console, jobs need to be either running or in the queue to get valid results. These applications attempt to simulate running and queued jobs. However, the speed of your hardware and the configuration of your environment might require these applications to be altered in order for jobs to be placed in waiting or captured while they are running.

C00_Suspend_Server.java

This sample suspends the Distribution Server. The server remains running, but it stops polling the WebFOCUS repository for scheduled jobs and will not run new jobs. If a job is submitted while the Distribution Server is suspended, a message is returned stating that the server has been suspended.

Usage: java C00_Suspend_Server

C01_Resume_Server.java

This sample resumes a suspended Distribution Server.

Usage: java C00_Resume_Server

C10_Get_InQ.java

This sample creates three schedules in an attempt to get them to run simultaneously, so that at least one is placed in the queue. It then queries the queue and shows a list of waiting jobs.

Usage: java C10_Get_InQ

C11_Get_Running.java

This sample creates three schedules in an attempt to get them to run simultaneously, so that at least one is running. It then queries the queue and shows a list of running jobs.

Usage: java C11_Get_Running

C31_Update_Priority.java

This sample runs two schedules in an attempt to place the second one on the queue. This application, if it can access the second job, will change its priority from the default (3) to whatever is designated in the command line. This accepts one argument: the new priority.

Usage: javaC31_Update_Priority <priority>

C32_Delete_FromQ.java

This sample runs two schedules in an attempt to place the second one on the queue. This application, if it can access the second job, will delete this job from the queue and then print the remaining jobs in the queue to standard out.

Usage: java C32_Delete_FromQ

Log Samples

The Log directory contains sample Java applications that show how to use the ReportCaster logging features in the API. As with all of the samples, it uses the ReportCaster authentication and context information contained in samples.properties. These sample Java applications are presented as examples for learning purposes and are not represented as production quality applications.

G10_Get_OwnerList.java

Writes a list of log owners to standard out.

Usage: java G10_Get_OwnerList

G11_Get_LogByJobId.java

This application does a number of things in an attempt to give examples of functionality. It accepts two arguments: a schedule owner and the Job Description of the schedule. Given the owner and the schedule description, it first runs the specified job. It then waits for the job to be complete and displays the log information.

Usage: java G11_Get_LogByJobId <owner> <scheduleDescription>

G12_Get_LastLog.java

This application gets the last log for a specified owner and job description and displays the information to standard out. It accepts two arguments: the schedule owner and the job description.

Usage: java G12_Get_LastLog <owner> <scheduleDescription>

G13_Get_LogListBySchedule.java

This application writes log information to standard out for all logs generated by the specified schedule. It accepts two arguments: the schedule owner and the job description.

Usage: java G13_Get_LogListBySchedule <owner> <scheduleDescription>

G14_Get_LogListByOwner.java

This application writes log information to standard out for all of the logs belonging to a particular owner. It accepts one argument: the owner of the schedules.

Usage: java G14_Get_LogListByOwner <owner>

G15_Get_LogListByOwnerByCalendar.java

This application writes log information to standard out for all of the logs belonging to a particular owner that ran on or after a specified date. It accepts two arguments: the owner of the schedule(s) and the start date.

Usage: java G15_Get_LogListByOwnerByCalendar <owner><startDate>

G30_Delete_LogListBySchedule.java

This application deleted all logs for a specific schedule. It accepts two arguments: the owner and Job Description.

Usage: java G30_Delete_LogListBySchedule <owner> <scheduleDescription>

Credential Authentication Samples

This section describes sample JSPs that show how to use the ReportCaster authentication functionality in the API. For the Java API, there are three out of the box implementations of the Credential interface: PasswordCredential, SecurityTokenCredential, and WebAuthenticateCredential.

FirstPage.jsp

This sample shows how the WebAuthenticateCredential class and the PasswordCredential class can be used together for single sign-on purposes. It tests to see if a remote user exists on the web server. If it does not, it prompts for a user name and password. Upon authentication, PasswordCredential will generate a cookie, which will allow the second page to function.

NextPage.jsp

This page shows authentication through cookies using the SecurityTokenCredential class.

Troubleshooting API Samples

When running the sample applications, you might encounter the following problems:

- The sample applications both create and delete scheduling objects. The first time the samples are run, the deleted applications will generate error messages because the objects they delete have not yet been created. They are created during the first run of the samples. On subsequent runs of the sample applications, the delete applications should run without error because the objects they delete now exist.
- You might also get errors when you run the console applications. This is because the console reports on schedules that are in the process of running or are waiting to run.

Depending on the speed and load of your ReportCaster Distribution Server, the sample schedules may no longer be running or on the queue when the sample applications are run, resulting in an error. Developers should feel free to alter the sleep intervals or substitute longer running reports if this behavior occurs.

ReportCaster API Subroutines

As of Version 4 Release 3.5, the ReportCaster Servlet API is functionally stabilized. For Version 5.3 and higher, including Version 7 Release 7, Information Builders recommends that you use the ReportCaster API to implement all ReportCaster functionality. In Release 8.0 Version 05, Information Builders recommends the use of RESTful web services to implement ReportCaster functionality.

This section is a reference for users who have already developed applications using ReportCaster API subroutines in a previous release.

API Subroutines

ReportCaster API subroutines are a set of C-based API functions that call a servlet. The subroutines are called from a procedure and 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.

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. For more information, see Maintaining Distribution Lists Using the DSTBULK Subroutine.
DSTMEM	Maintains single members in a Distribution List. Enables the user to add a new member or delete an existing member. For more information, see Maintaining Single Distribution List Members Using the DSTMEM Subroutine.

API Subroutine	Description
DSTRUN	Immediately runs and distributes a report. For more information, see Running a Scheduled Job Using the DSTRUN Subroutine.

Subroutine Security

All ReportCaster API subroutines require an Execution ID and password as the first and third arguments. The subroutines use this user ID and password internally to perform a ReportCaster sign-in.

ReportCaster uses the Execution ID to determine ownership of schedules, Distribution Lists, and log files. Your Execution ID is stored in the WebFOCUS repository in the case you specify at sign-in. You must specify your Execution 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.

Note: Internally, the DSTBULK subroutine calls the DSTDLBULK servlet. The DSTRUN subroutine calls the DSTRUNNOW servlet, while the DSTMEM subroutine calls the DSTDLMEM servlet.

Other security considerations are:

- The DSTBULK and DSTMEM subroutines require that an owner 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 user schedules. When running another user schedule, the owner user ID must be supplied. This ensures that the user has knowledge of the owner 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.
- Procedure called by an HTML form.
- 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 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 the *ibi™ WebFOCUS® Developing Reporting Applications* manual.

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/pswd',httpuser/pswd_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 Execution ID. It must match the user ID stored in the WebFOCUS repository with the job information.

srv_userid_length

Is the length of the Execution ID.

srv_userpass

Is a valid password for the Execution ID. It must match the password of the user ID stored in the WebFOCUS repository with the job information.

srv_userpass_length

Is the length of the password for the Execution ID.

hostname_port

Is the host name (or IP address) and port number of the web server on which the WebFOCUS Client is installed. The maximum character length is 30.

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. The maximum character length is 10. For example,

```
'hostname:81',
```

hostname_port_length

Is the length of the host name and port number. This is 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 forward slash 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 forward slash 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,** which resolves special connectivity problems.
- **0,** which indicates that there are no special connectivity problems.

'I4'

Is the format of the return code. For more information, see ReportCaster API Subroutine Messages.

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 Call an API Subroutine Using -SET. The arguments are positional.

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

The only exception is the argument *tcpiplevel*, 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.

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
  ```

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 Execution 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 WebFOCUS 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.

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 the *ibi™ 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 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 preprocessing procedure of the schedule being submitted.

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.

srv_userid Argument

Description	Valid Execution ID. It must match the user ID stored in the WebFOCUS repository.
Required?	Yes
Format	A48 (Windows 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

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

srv_userpass Argument

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

srv_userpass_length Argument

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

hostname_port Argument

Description	Is the connection information to the web or application server where the WebFOCUS web application is deployed. The connection information includes the host name, port number, web context root information for the WebFOCUS web application, and the ReportCaster web context root. For example: `'hostname:port/webfocus_context_root:rcaster_`

	`context_root'` - *hostname* (or IP address) and *port* number of the web server on which the WebFOCUS Client is installed. This must be separated by a colon. The maximum character length is 30. 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. The maximum port number character length is 10. For example, *hostname*:81. - *webfocus_context_root* is the WebFOCUS web application context root. The default is ibi_apps. The maximum character length for the context root is 40. - *rcaster_context_root* is required to be the same value as the *webfocus_context_root*. The maximum character length for the context root is 40. **Note:** Hard coded references to the ReportCaster rcaster context root or a ReportCaster specific context root in ReportCaster Subroutine API calls or Web application URLs are not supported in Release 8. For more information, contact Customer Support.
Required?	Yes
Format	A64
Valid Values	No restrictions.

hostname_port_length Argument

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

Required?	Yes
Format	I4
Valid Values	Integer

name Argument

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

name_length Argument

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

function Argument

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

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

access Argument

Description	Length of the value of the function.
Required?	Yes, if *function* is C. Ignored if *function* 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

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

method Argument

Description	Distribution method for a new Distribution List.
Required?	Yes, if *function* is C. Ignored if *function* is N, R, or D.
Format	A5
Valid Values	FTP or F = FTP distribution.MAIL or M = email distribution. PRINT or P = Printer distribution.

method_length Argument

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

user Argument

Description	One of the following: - User ID of the owner of the target Distribution List when *copy* 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 *copy* is present. The owner of the target Distribution List is the Execution ID, passed as the first argument to the subroutine.
Required?	Yes This feature enables a sign-in user to create or manipulate a Distribution List for another user, whose ID is supplied on *owner*.
Format	A48 (Windows 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

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

filename Argument

Description	Name of the external source file used to supply member names. The following requirements apply: • The file must be located in EDAPATH (Windows and UNIX), or in a partitioned data set (PDS) allocated to ddname EDARPC (OS/390). • The file must have the extension .fex on Windows and UNIX, although you do not specify it. **Note:** This is similar to the way the IBIB_filename parameter is used for DSTDLBULK. For more information, see Maintaining Distribution Lists Using the DSTDLBULK Servlet.
Required?	Yes, if supplying members from an external file when *function* is C, N, or R.
Format	A8
Valid Values	First character must be alphanumeric. Single quotation marks and ampersands are not allowed. Blank lines are not allowed in an external distribution file.

filename_length Argument

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

copy Argument

Description	Name of the source Distribution List. Applies when *function* is one of the following: • 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

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

httpuser/pswd Argument

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

Description	Length of the web server user ID and password, including the forward slash character (/).
Required?	Yes, if *httpuser/pswd* is supplied.
Format	I4
Valid Values	Integer

tcpiplevel Argument

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

Description	Value that DSTBULK returns to the calling procedure, indicating successful completion (0) or an error condition. For more information, see ReportCaster API Subroutine Messages.
Required?	Yes
Format	I4
Valid Values	'I4'

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. The numbers to the left of the code refer to the annotations that follow.

```
1.  FILEDEF SUBLOG DISK D:\ibi\srv77\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 writes error codes and messages (if there are any) to this file, as coded in line 14. The -RUN command executes the FILEDEF command.

   For more information about the FILEDEF command, see the *ibi™ 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 Execution ID ('userid'), included in single quotation marks, and length of 'userid' (6), are the first arguments passed to the subroutine.

3. The Execution ID 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 ReportCaster API Subroutine Messages.

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.

Maintain Single Distribution List Members Using DSTMEM

```
-SET &var_name=DSTMEM(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.

DSTMEM 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.

srv_userid Argument

Description	Valid Execution ID.

Required?	Yes
Format	A48 (Windows 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

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

srv_userpass Argument

Description	Valid password for the Execution ID.
Required?	Yes
Format	A48
Valid Values	Alphanumeric characters and spaces are allowed.

srv_userpass_length Argument

Description	Length of the password for the Execution ID.
Required?	Yes
Format	I4
Valid Values	Integer

hostname_port Argument

Description	Is the connection information to the web or application server where the WebFOCUS web application is deployed. The connection information includes the host name, port number, web context root information for the WebFOCUS web application, and the ReportCaster web context root. For example: `'hostname:port/webfocus_context_root:rcaster_context_root'` • *hostname* (or IP address) and *port* number of the web server on which the WebFOCUS Client is installed. This must be separated by a colon. The maximum character length is 30. 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. The maximum port number character length is 10. For example, *hostname*:81. • *webfocus_context_root* is the WebFOCUS web application context root. The default is ibi_apps. The maximum character length for the context root is 40.

	- *rcaster_context_root* is required to be the same value as the *webfocus_context_root*. The maximum character length for the context root is 40. **Note:** Hard coded references to the ReportCaster rcaster context root or a ReportCaster specific context root in ReportCaster Subroutine API calls or Web application URLs are not supported in Release 8. For more information, contact Customer Support.
Required?	Yes
Format	A64
Valid Values	No restrictions.

hostname_port_length Argument

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

name Argument

Description	Name of the Distribution List that will be modified.
Required?	Yes

Format	A50
Valid Values	First character must be alphabetic.

name_length Argument

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

function Argument

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

Description	Length of the value of the function.
Required?	Yes

Format	I4
Valid Values	1

owner Argument

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

user_length Argument

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

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

Description	Length of the burst value.
Required?	Yes, if *fldvlu* is supplied.
Format	I4
Valid Values	I4

destfn Argument

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

destfn_length Argument

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

httpuser/pswd Argument

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

Description	Length of the web server user ID and password, including the forward slash character (/).
Required?	Yes, if *httpuser/pswd* is supplied.
Format	I4
Valid Values	Integer

tcpiplevel Argument

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

Description	Value that DSTMEM returns to the calling procedure, indicating successful completion (0) or an error condition.
Required?	Yes
Format	I4
Valid Values	'I4'

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. The numbers to the left of the code refer to the annotations that follow.

```
1. FILEDEF SUBLOG DISK D:\ibi\srv77\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 writes error codes and messages (if there are any) to this file, as coded in line 12. The -RUN command executes the FILEDEF command.

   For more information about the FILEDEF command, see the *ibi™ 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 Execution ID ('userid'), included in single quotation marks, and length of 'userid' (6), are the first arguments passed to the subroutine.

3. The Execution ID 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 email address (&VAR3) and the length of the email 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 this 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 ReportCaster API Subroutine Messages.

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.

Note: To run the scheduled job with an execution ID other than the execution ID stored with the schedule information in the ReportCaster tables, see Running a Schedule With a Different Execution ID

DSTRUN Requirements

- As of WebFOCUS Release 8.0 Version 08, you can identify a schedule using DSTRUN as follows:
    - For a migrated schedule:
        - You can use the description or the schedule ID.
    - For a new schedule:
        - You can use the full IBFS path and file name, instead of the Description/Title, which can be found by viewing the Properties of the Schedule.
        - You can use the Schedule ID, which can be found by right-clicking the Schedule and selecting **View Log**.
- The call must be made from a secure environment. This means that the user must log on to your application with a valid Execution ID and password. The user ID and password for the server must match the user ID and password stored in the WebFOCUS repository with the target job schedule.
- To run another user job, you must supply the user 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*.

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,
[- vtcpiplevel,']
-  '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.

srv_userid Argument

Description	Valid user ID on WebFOCUS Reporting Server and ReportCaster user with schedule privilege. It must match the user ID stored in the WebFOCUS repository.
Required?	Yes

Format	A48 (Windows 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

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

srv_userpass Argument

Description	Valid password for srv_userid. Must be the same value to validate password on WebFOCUS Reporting Server and ReportCaster table.
Required?	Yes
Format	A240
Valid Values	Alphanumeric characters and spaces are allowed.

srv_userpass_length Argument

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

hostname_port Argument

Description	Is the connection information to the web or application server where the WebFOCUS web application is deployed. The connection information includes the host name, port number, web context root information for the WebFOCUS web application, and the ReportCaster web context root. For example:
	`'hostname:port/webfocus_context_root:rcaster_context_`
	`root'`
	• *hostname* (or IP address) and *port* number of the web server on which the WebFOCUS Client is installed. This must be separated by a colon. The maximum character length is 30.
	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. The maximum port number character length is 10. For example, *hostname*:81.
	• *webfocus_context_root* is the WebFOCUS web application context root. The default is ibi_apps. The maximum character length for the context root is 40.
	• *rcaster_context_root* is required to be the same value as the

	webfocus_context_root. The maximum character length for the context root is 40. **Note:** Hard coded references to the ReportCaster rcaster context root or a ReportCaster specific context root in ReportCaster Subroutine API calls or Web application URLs are not supported in Release 8. For more information, contact Customer Support.
Required?	Yes
Format	A123
Valid Values	No restrictions.

hostname_port_length Argument

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

scheduleid Argument

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 *jobdesc* is required.

Format	A12
Valid Values	First character must be alphanumeric. Single quotation marks and ampersands are not allowed.

scheduleid_length Argument

Description	Length of *scheduleid*.
Required?	Yes, if *scheduleid* is supplied.
Format	I4
Valid Values	Integer

jobdesc Argument

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

jobdesc_length Argument

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

priority Argument

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

Description	User ID of the owner of the scheduled job. Optionally, an execution ID other than the execution ID stored with the schedule information in the ReportCaster tables can be specified in the schedule_owner parameter. The specified execution ID must conform to the security rules for the ReportCaster Subroutine API. For more information, see Running a Schedule With a Different Execution ID.

Required?	Yes. This feature enables a sign-in user to run a job owned by another user, whose ID is supplied on *owner*.
Format	A99 (Windows 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

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

parm Argument

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	A151
Valid Values	Must fit on an 80-byte line, which includes all punctuation.

parm_length Argument

Description	Length of the parameter string.
Required?	Yes, if *parm* 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

Description	Web server user ID and password.
Required?	Yes, if the web server is running with security on.
Format	A1000
Valid Values	Valid web server user ID and password.

httpuser/pswd_length Argument

Description	Length of the web server user ID and password, including the forward slash character (/).
Required?	Yes, if *httpuser/pswd* is supplied.
Format	I4
Valid Values	Integer

tcpiplevel Argument

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

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

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. The numbers to the left of the code refer to the annotations that follow.

```
1.  FILEDEF SUBLOG DISK D:\ibi\srv77\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 writes error codes and messages (if there are any) to this file, as coded in line 12. The -RUN command executes the FILEDEF command.

   For more information about the FILEDEF command, see the *ibi™ 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 Execution ID ('userid'), included in single quotation marks, and length of 'userid' (6), are the first arguments passed to the subroutine.

3. The Execution ID password ('mypass') and length of the password (6) are passed to the subroutine.

4. The name of the web server or application server running the WebFOCUS Client 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 ID length 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 this 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 forward slash 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 ReportCaster API Subroutine Messages.

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

Running a Schedule With a Different Execution ID

To run the scheduled job with an execution ID other than the execution ID stored with the schedule information in the ReportCaster tables, you can specify the execution ID within the schedule_owner parameter in the DSTRUN subroutine call. The maximum size of the schedule_owner parameter is 200 characters.

The specified execution ID must conform to the security rules for the ReportCaster Subroutine API which means that it must be a valid user ID and password on the WebFOCUS Reporting Server, and a ReportCaster user with schedule privilege.

The format to specify a different execution ID in the schedule_owner parameter is *schedule_owner:execID*. In the example below, the execution ID is specified in line 10. The execution ID specified in the schedule_owner parameter must match the execution ID specified the execID parameter in line 4.

```
1.  FILEDEF SUBLOG DISK C:\SUBLOG.FTM
2.  RUN
3.  -SET &SUBERR=DSTRUN(
4.  - 'execid2',7,
5.  - 'execid2',7,
6.  - 'jetsetter:8080',14,
7.  - 'S11o9sddut01',12,
8.  - ' ',0,
9.  - 5,
10. - 'admin:execid2',13,
11. - ' ',0,
12. - ' ',0,
13. - 'I4');
14.  -IF &SUBERR EQ 0 GOTO EXIT;
    -INCLUDE DSTRUNER
    -WRITE SUBLOG &SUBERR
    -EXIT
```

If the execution ID specified in the schedule_owner parameter does not match the execution ID specified in the execID parameter, the execution ID value stored for the schedule in the ReportCaster tables is used to run the scheduled job.

If the execution ID and password in the WF_COOKIE exists in the ReportCaster tables (BOTUPROF) but the password in the WF_COOKIE does not match the password in the ReportCaster tables, an authentication error is given.

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.

Using Amper Variables for the Burst Value and Email 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\srv77\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
```

ReportCaster API Subroutine Messages

A ReportCaster API subroutine called by a procedure provides a return code that indicates whether or not a subroutine was 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 following message: The password is not valid.

On Windows, 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 the *ibi™ 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 WebFOCUS 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.

DSTBLKER (DSTBULK Return Codes)

The translation procedure DSTBLKER includes all return codes and messages for the DSTBULK subroutine.

ReportCaster Servlet API

This section provides information on the functionally stabilized ReportCaster Servlet API.

Overview

As of Version 4 Release 3.5, the ReportCaster Servlet API is functionally stabilized. For Version 5.3 and higher, including Version 7 Release 7, Information Builders recommends that you use the ReportCaster API to implement all ReportCaster functionality. In Release 8.0 Version 05, Information Builders recommends the use of RESTful web services to implement ReportCaster functionality.

This chapter is a reference for users who have already developed applications using the Servlet API in a previous release.

ReportCaster Servlet API Descriptions

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.

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.

Servlet	Description
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.

Servlet Security

Before a user can access a ReportCaster Servlet API application, a valid WebFOCUS Reporting Server sign-in must take place. This sign-in is typically accomplished with an HTML sign-in form.

Note: To create a schedule with the Servlet API, a valid Execution ID is required for the default ReportCaster Reporting Server.

Creating Cookies for User Credentials

WebFOCUS cookies are established when a valid ReportCaster sign-in occurs using the WF_SIGNON action.

You can create your own custom form that assigns the value WF_SIGNON to the variable IBIWF_action. This causes the creation of the cookie containing the validated server credentials.

Determining Ownership of Schedules and Distribution Lists

There are three options for assigning ownership with the ReportCaster Servlet API:

- Default User. If there is a value for the Default User parameter in the ReportCaster Server Configuration tool, this value will be used as the owner of all migrated schedules and Distribution Lists. For more information, see Servlet Security.
- IBIB_user. If this parameter is explicitly set on the HTML form calling the servlet, it will be used as the owner ID of schedules and Distribution Lists. For more information, see Servlet Security.
- IBIC_user. If IBIB_user is not supplied on the calling form, then IBIC_user (from the WebFOCUS cookie) is assigned as the owner of schedules and Distribution Lists.

Determining Execution Credentials for Schedules

Execution credentials are used at schedule execution time to log on to the default WebFOCUS Reporting Server and execute the job. Execution credentials are taken from the WebFOCUS cookie that is created when the user signs on to the ReportCaster Servlet API application. The WebFOCUS cookie may include credentials for multiple servers. However, Servlet API schedules can only run on the default WebFOCUS Reporting Server.

The ReportCaster Servlet API determines which WebFOCUS cookie credentials to use for validation as follows:

1. Search for the default server as defined in the ReportCaster Server Configuration tool and use the credentials associated with that server.
2. If the default server from the ReportCaster Distribution Server configuration file is not found, look for * in the WebFOCUS cookie, indicating that the sign-in form did not specify a server. Attempt to validate the user ID with the credentials associated with this non-specific server.
3. If the default server and * are not located in the WebFOCUS cookie, an error message is issued.

Default User in Configuration Tool

The Default User is set using the ReportCaster Server Configuration tool. If there is a value for the Default User parameter, this value is assigned as the owner of API-based schedules that are migrated from an earlier release. This value is also assigned as the owner of new schedules and Distribution Lists created by the ReportCaster Servlet API.

If the Default User parameter is left blank in the configuration tool, ownership of migrated schedules is assigned to the Execution ID from the schedule. Leaving the Default User blank allows schedules and Distribution Lists to be distinguished as belonging to different users. Execution IDs must exist as valid ReportCaster users in the appropriate tables. Execution IDs can be created as ReportCaster users with the Managed Reporting User Administration tool, or they can be migrated and created as ReportCaster users with the migration tool.

IBIB_user Parameter

Using the ReportCaster Servlet API, it is possible to run a schedule and maintain a Distribution List belonging to an owner other than the signed in user. To accomplish this, you must pass the IBIB_user parameter (through an HTML sign-in form) to the Servlet API as the alternate owner.

Note: If the ReportCaster user ID used to sign in to the Servlet API application is a ReportCaster Administrator user ID, the IBIB_user parameter is ignored and access is provided to all schedules and Distribution Lists in the WebFOCUS repository.

The following table describes the use of the IBIB_user parameter by the servlets.

Servlet	IBIB_user Description	Notes
DSTDLBULK	Required if the user value in the WebFOCUS cookie is not the owner of the Distribution List, and the user is adding members to an existing Distribution List, replacing members in	The owner of the scheduled job is the user ID from WF_ COOKIE.

Servlet	IBIB_user Description	Notes
	an existing Distribution List, or deleting an existing Distribution List. The owner of the scheduled job is the user ID from WF_COOKIE. For the copy function, IBIB_user can be supplied as an alternate owner ID for the target list.	
	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	Not applicable.	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. Optionally, an alternate execution ID can be supplied.	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

Servlet	IBIB_user Description	Notes
		through the ReportCaster API and are owned by the current WebFOCUS Reporting Server user ID.

Calling a Servlet From an HTML Form

You can call a servlet by either:

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

Call a Servlet From an HTML Form

Specifying an Absolute Address:

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

where:

context_root

Is the configurable context root for the ReportCaster web application in your Application Server configuration. By default, this is /ibi_apps.

hostname

Is the host name of the web server.

servlet_name

Is the name of the servlet.

Specifying a Relative Address:

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

where:

context_root

Is the configurable context root for the ReportCaster web application in your Application Server configuration. By default, this is /ibi_apps.

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 uses the default.
- If you do not supply a required value and no default exists, the servlet returns a message. For example, 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.
```

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 Execution ID. The description can be up to 90 alphanumeric characters with embedded blanks and special characters.
- **IBIB_scheduleid.** This is a unique, API-generated ID that identifies a scheduled job in the WebFOCUS 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.

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. See Maintaining Distribution Lists Using the DSTDLBULK Servlet.
- DSTDLMEM to maintain single Distribution List members. See Maintaining Single Distribution List Members Using the DSTDLMEM Servlet.
- DSTDLLIST to display a Distribution List. See Displaying a Distribution List Using the DSTDLLIST Servlet.

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.

Specifying an Absolute Address Using DSTDLBULK

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

where:

context_root

Is the configurable context root for the ReportCaster web application in your Application Server configuration. By default, this is /ibi_apps.

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

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

Description	Name of source Distribution List. Applies when IBIB_function is one of the following:

	• 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

Description	Name of external source file used to supply member names. The file must: • Reside on EDAPATH (Windows and UNIX), or in a partitioned data set (PDS) allocated to ddname EDARPC (OS/390). • Have the extension .fex on Windows and UNIX, although you do not specify it. For more information about IBIB_filename, see Maintaining Distribution Lists Using the DSTDLBULK Servlet.
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

Description	Type of maintenance action that will be performed.
Required?	Yes
Size in bytes	1
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.
Default	C

IBIB_method Parameter

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

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

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), email address, or printer name. For example: File name ENGLAND.HTM,,$Email address Alfred_Stevens@abcd.com,,$Printer name \\WERIBI\29B2.PRNTNY.IBI,,$
Default	None

IBIB_tcpiplevel Parameter

Description	Method ReportCaster uses for securing sockets from TCP/IP,

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_user Parameter

Description	One of the following: • 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 target 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?	**For adding, replacing, or deleting:** A value is required if IBIB_function is N, R, or D. Only public Distribution Lists may be copied. **For creating:** Only a ReportCaster Administrator can create a Distribution List for another user. This must be a public Distribution List.
Size in bytes	48 (Windows 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-burst report distribution, and an additional burst value field for burst 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.

Note: Email addresses and domains added to an external Distribution List will be validated against the Allowed Email Domains and Addresses list in the ReportCaster Console. For more information, see the *ibi™ WebFOCUS® ReportCaster Guide* technical content.

Creating an External File for a Burst Report

For a burst 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 burst 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 email 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,$
```

Creating an External File for a Non-Burst Report

For a non-burst 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 email address for email 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 burst 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-burst 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.

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.

Specifying an Absolute Address Using DSTDLMEM

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

where:

context_root

Is the configurable context root for the ReportCaster web application in your Application Server configuration. By default, this is /ibi_apps.

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

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

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

IBIB_name Parameter

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

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_user Parameter

Description	User ID of the owner of the Distribution List being added to or deleted from.
Required?	Yes
Size in bytes	48 (Windows 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

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.

Specifying a Relative Address Using DSTDLLIST

```
<FORM ACTION="/context_root/servlet/DSTLIST">
```

where:

context_root

Is the configurable context root for the ReportCaster web application in your Application Server configuration. By default, this is /ibi_apps.

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

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

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.

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 the 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 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.

Specifying a Relative Address Using DSTSCHED

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

where:

context_root

Is the configurable context root for the ReportCaster web application in your Application Server configuration. By default, this is /ibi_apps.

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

Description	Status of job.

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

IBIB_byfield Parameter

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

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

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

IBIB_enddate Parameter

Description	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.
Required?	No

Size in bytes	8
Valid Values	A date specified as *YYYYMMDD*.
Default	20990101

IBIB_endtime Parameter

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

IBIB_frequency Parameter

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

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

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

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 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

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

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

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 Scheduling a Job Using the DSTSCHED Servlet.
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

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

Description	Name of a procedure and optionally its parameter string that will run prior to the scheduled report request. This procedure performs setup tasks. An example of an IBIB_prerpc1 value is: 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

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

Description	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.
Required?	No
Size in bytes	8
Valid Values	A date specified as *YYYYMMDD*.
Default	Current system date.

IBIB_starttime Parameter

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

IBIB_tcpiplevel Parameter

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

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

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

Description	Email 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

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 Server configuration file.

IBIB_mailsubject Parameter

Description	Text that describes the content or purpose of the email.
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

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

IBIB_ftphost Parameter

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 Server configuration file.

IBIB_ftplocation Parameter

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 Server configuration file.

IBIB_ftppass Parameter

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 Server configuration file.

IBIB_ftpuser Parameter

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 Server configuration file.

Notification Parameters for DSTSCHED

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

IBIB_notifyaddress Parameter

Description	Email 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

Description	Email 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

Description	Email 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

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 WebFOCUS 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 (=).

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 (').

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.

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_TITLE" value="DOG DAY AFTERNOON"
```

The query string assignment that results is:

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

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 (').

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 signed in 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.

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.

Specifying a Relative Address Using DSTACTIVE

```
<FORM ACTION="/context_root/servlet/DSTACTIVE">
```

where:

context_root

Is the configurable context root for the ReportCaster web application in your Application Server configuration. By default, this is /ibi_apps.

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

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

IBIB_jobdesc Parameter

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	• Entering ALL updates the status of all jobs in BOTSCHED (requires ReportCaster Administrator user ID). or • An existing description used for the schedule created by

	the ReportCaster API.
Default	None

IBIB_scheduleid Parameter

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	• Entering ALL updates the status of all jobs in BOTSCHED (requires a ReportCaster Administrator user ID). or • An existing schedule ID generated by the ReportCaster API.
Default	None

IBIB_tcpiplevel Parameter

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 WebFOCUS 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 Scheduling a Job Using the DSTSCHED Servlet, making sure to substitute DSTRUNNOW for every instance of DSTSCHED.

Note: To run the scheduled job with an execution ID other than the execution ID stored within the schedule information in the ReportCaster tables, append a colon and the alternate execution ID to *schedule owner* in the IBIB_user parameter in the URL query.

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 Execution 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 WebFOCUS repository for the target job schedule.

- To run a job belonging to another user, supply their user ID for the IBIB_user 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.

Specifying a Relative Address Using DSTRUNNOW

```
<FORM ACTION="/context_root/servlet/DSTRUNNOW">
```

where:

context_root

Is the configurable context root for the ReportCaster web application in your Application Server configuration. By default, this is /ibi_apps.

Specifying an Alternate Execution ID Using DSTRUNNOW

To specify an execution ID other than the execution ID stored with the schedule information in the ReportCaster tables, append a colon and the alternate execution ID to the schedule owner in the IBIB_user parameter. For example, in the first request, the default execution ID runs the scheduled job:

```
http://hostname/context_root/servlet/DSTRUNNOW?IBIB_user=admin...
```

In the following request, *execid* is specified as the execution ID:

```
http://hostname/context_root/servlet/DSTRUNNOW?IBIB_user=admin:execid...
```

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

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 the schedule created by the ReportCaster API.
Default	None

IBIB_parm Parameter

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 Scheduling a Job Using the DSTSCHED Servlet.
Required?	No
Size in bytes	75
Valid Values	Must fit on an 80-byte line, which includes all punctuation.
Default	None

IBIB_priority Parameter

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

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

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_user Parameter

Description	User ID of the owner of the scheduled job. Optionally, an execution ID can also be supplied.
Required?	No. If you do not supply a value, DSTRUNNOW uses the user ID from the WebFOCUS cookie and execution ID stored with the schedule information in the ReportCaster tables. If you do supply an owner value, it will override the user ID from the WebFOCUS cookie and ReportCaster tables. This feature enables a signed in user (whose ID is stored in the cookie) to run a job owned by another user. In addition, if an execution ID is also supplied, it will override the execution ID stored with the schedule if the specified execution ID meets the Servlet security requirements.
Size in bytes	48 (Windows 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 WebFOCUS 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 appears:

```
No Log information exists for this request.
```

Specifying a Relative Address Using DSTLOG

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

where:

context_root

Is the configurable context root for the ReportCaster web application in your Application Server configuration. By default, this is /ibi_apps.

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

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

IBIB_endtime Parameter

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

IBIB_jobdesc Parameter

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

IBIB_lastexec Parameter

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

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

IBIB_startdate Parameter

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

IBIB_starttime Parameter

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

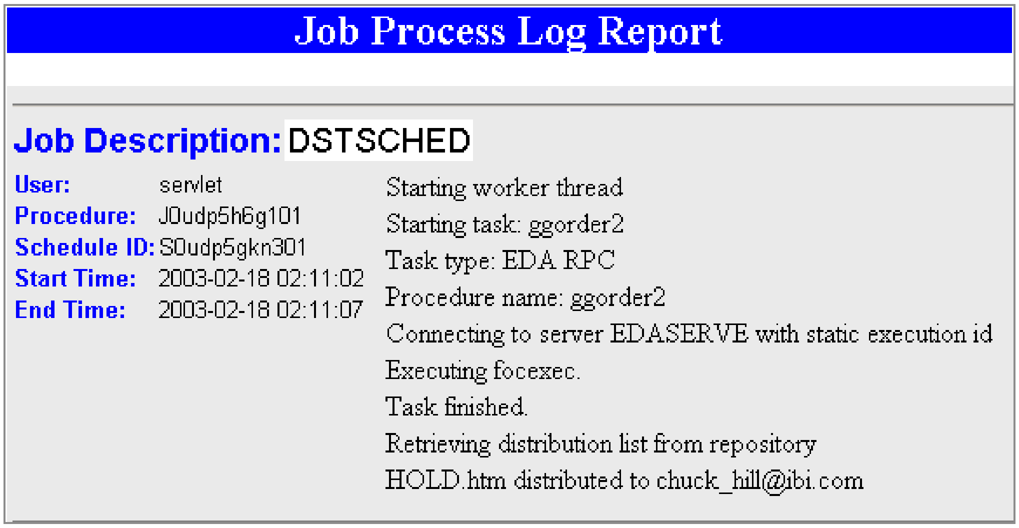
IBIB_tcpiplevel Parameter

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

Reading a Log Report

The Job Process Log Report displays information according to your specifications in a separate browser window. One log record is produced for each scheduled job run in the specified time frame.

The following image shows a typical Job Process Log Report.



Job Process Log Report

Job Description: DSTSCHED

User: servlet
Procedure: J0udp5h6g101
Schedule ID: S0udp5gkn301
Start Time: 2003-02-18 02:11:02
End Time: 2003-02-18 02:11:07

```
Starting worker thread
Starting task: ggorder2
Task type: EDA RPC
Procedure name: ggorder2
Connecting to server EDASERVE with static execution id
Executing focexec.
Task finished.
Retrieving distribution list from repository
HOLD.htm distributed to chuck_hill@ibi.com
```

A Job Process Log Report includes the following information:

- **Job Description.** Unique description that you supplied to identify the schedule.
- **User.** ReportCaster user ID, indicating the owner of the schedule.
- **Procedure.** Unique, ReportCaster-generated key that identifies a specific execution of a scheduled job.
- **Schedule ID.** Unique, ReportCaster-generated key assigned to the job when it was scheduled.
- **Start Time.** Date and time the job started running.
- **End Time.** 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 was completed. Processing information also includes reasons why a request failed, such as the unavailability of a data source.

Glossary

This section provides definitions for ReportCaster specific terms as well as security and Java concepts.

Address Book

The interface for creating a distribution list for a ReportCaster schedule.

API/SQL (Application Programming Interface for Structured Query Language)

A component of the server product from Information Builders. API/SQL is a call-level interface that enables third-generation language application programs to manage local and remote databases.

Automatic Passthru (APT)

The ability of the Hub or Full-function Server to detect that the SQL received is compatible with the target engine and so does not need to be translated.

Beans

Components (objects) written by a Java developer that extend JavaBeans components (for example, DSTSchedFactory).

BLOB

Binary Large Object.

burst value

The filter used for breaking a report into relevant sections.

bursting

A feature that breaks a report into sections to be distributed separately. Bursting enables specified sections of a report to be distributed to individual recipients.

Catalog

A registry of data that describes objects used by SQL applications. Similar to the native catalog of a RDBSM, an SQL server has its own catalog for keeping track of data sources, stored procedures, and other objects.

SQL catalogs include:

Dynamic Catalog. A registry of data about EDA/SQL data sources, columns in data sources, collections of data sources, stored procedures, and user IDs. It resides on a Hub or Full-function Server.

Relational Gateway Catalog. An optional extension to the native RDBMS catalog when using a Relational Gateway. This catalog tracks data sources, columns, keys, indexes, collections, stored procedures, creators, and supported data types. It is necessary when using the EDA/Extender for Microsoft ODBC, EDA/Compose, or EDA/Extender for Lotus DataLens. It is also required by any client application that uses metadata about stored procedures or runs stored procedures written to access EDA/SQL metadata.

class

The formal description of an object. The class acts as the template from which an instance of an object is created at run time. The class defines the properties (member variables) of the object and the member functions used to control the object's behavior. (Source: Microsoft online glossary).

CLOB

Character Large Object.

Custom Reports

A report that is created and edited using Report Assistant, Graph Assistant, or the Editor. Custom Reports are located in the Custom Reports folder located under the My Reports tab in the Managed Reporting Domains environment. Custom Reports are available to users who have been granted the Advanced privilege in Managed Reporting. Administrators automatically have the Advanced privilege and they can assign this privilege to Analytical Users and Developers.

Database Administrator

Is responsible for the maintenance and creation of databases and the maintenance of the database catalog or data dictionary.

Direct Passthru

The ability to send dialect-specific SQL, rather than EDA/SQL, to a relational server.

Distribution List

Defines the distribution method, Access List, burst value, and address for a specific job. Distribution Lists may be reused for multiple jobs.

distribution method

The means by which ReportCaster output is distributed. The distribution method may be by email, printer, or FTP.

Distribution Object

A Java object that includes the distribution list and the burst value, if any.

Distribution Server

A Java application that governs the process of submitting and distributing a scheduled job. The ReportCaster Distribution Server can be installed on the same platform as the WebFOCUS Reporting Server and the WebFOCUS components or it can be installed on a different platform.

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.

firewall

A system or combination of systems that enforces a boundary between two or more networks and keeps intruders out of private networks. Firewalls serve as virtual barriers to passing packets from one network to another. (Source: Microsoft online glossary).

inheritance

Generally, the ability of a newly created object to automatically have, or inherit, properties of an existing object. For example, a newly created child directory can inherit the access-control settings of the parent directory. A programming technique that duplicates the characteristics down a hierarchy from one class to another. (Source: Microsoft online glossary).

instance

An object of a particular class. Each instance has its own private data elements or member variables. See segment instance. (Source: Microsoft online glossary).

instantiate

To create an instance of an class. The process of creating or activating an object based on its class. (Source: Microsoft online glossary).

Java applet

An HTML-based program built with Java that a browser temporarily downloads to the hard disk of a user, from which location it runs when the web page is open. (Source: Microsoft online glossary).

JavaBeans

An object-oriented programming interface from Sun Microsystems that enables you to build reusable applications.

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).

JavaScript

A scripting language that evolved from Netscape's LiveScript language and was made more compatible with Java. It uses an HTML page as its interface. See script. (Source: Microsoft online glossary).

Library Access List

Defines the users and groups that are allowed to view the output of specified schedules distributed to the Report Library. Once the Access List is created, it can be used as often as needed. Each Access List is created as a private list that is known only by ReportCaster Administrators and the user who created it.

Managed Reporting Domain

Domains are the highest level of organization. Domains provide data on a particular topic (such as sales, inventory, or personnel). The data is stored in different forms in the following domain components: predefined reports (Standard Reports), data sources used to create reports (Reporting Objects), and reports created and saved by users (My Reports, Custom Reports, and Shared Reports).

method

A group of programming statements that perform a specific task (for example, getScheduleListHandler). Provides the means by which a user or application can utilize the services of an object. The methods available for an object are defined by the interface of the object.

My Reports

A My Report is a personal report that you save while working in a domain. Once you access a Reporting Object and create a report, you can save the report as a My Report. Once saved, you can run or edit these reports. No other user has access to your reports.

objects

The building blocks of object-oriented programming, an object is a self-contained module of data and its associated processes.

owner

The originator of a schedule. The owner has special security privileges, and is the only user, along with the administrator, to have access to a schedule if security is set to private.

package

A group of related classes and subpackages.

Procedure

A set of related control statements that cause one or more programs to be performed.

properties

Attributes of a method that affect object processing (for example, IBIB_scheduleid).

Report Library

An optional ReportCaster product that enables you to store and manage content distributed by ReportCaster for rapid retrieval and future use.

ReportCaster API

The ReportCaster Bean and Web Services API include full scheduling, log, Address Book, and Report Library capability for self-service users. The ReportCaster Bean and Web Services API is fully upward compatible with the ReportCaster Bean, Servlet, and Subroutine APIs, and is partially backward compatible.

ReportCaster Console

A set of HTML pages and JavaServer Pages (JSP) that enable ReportCaster Administrators to maintain and view schedule and log information, globally change values stored in the WebFOCUS repository, and manage the ReportCaster Distribution Server processing of schedules for execution.

Reporting Objects

A tailored view of a set of data that a Managed Reporting Administrator creates and saves to a group folder. The data contained in a Reporting Object is used to create personal reports quickly and in compliance with your company's reporting rules and guidelines.

Standard Report

A predefined procedure that an Administrator creates and stores in a group folder or subgroup folder. Standard Reports are used to retrieve data that changes on a regular basis. For example, monthly inventory reports or weekly sales reports. Each time you run a Standard Report, the output reflects the most current data while the format of the report remains constant.

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.

WebFOCUS Reporting Server

A server that processes a scheduled request, retrieves the data, and returns the report to the ReportCaster Distribution Server, which distributes the output.

ibi Documentation and Support Services

For information about this product, you can read the documentation, contact Support, and join Community.

How to Access ibi Documentation

Documentation for ibi products is available on the Product Documentation website, mainly in HTML and PDF formats.

The Product Documentation website is updated frequently and is more current than any other documentation included with the product.

Product-Specific Documentation

The documentation for this product is available on the ibi™ WebFOCUS® Documentation page.

How to Contact Support for ibi Products

You can contact the Support team in the following ways:

- To access the Support Knowledge Base and getting personalized content about products you are interested in, visit our product Support website.
- To create a Support case, you must have a valid maintenance or support contract with a Cloud Software Group entity. You also need a username and password to log in to the product Support website. If you do not have a username, you can request one by clicking **Register** on the website.

How to Join ibi Community

ibi Community is the official channel for ibi customers, partners, and employee subject matter experts to share and access their collective experience. ibi Community offers access to Q&A forums, product wikis, and best practices. It also offers access to extensions, adapters, solution accelerators, and tools that extend and enable customers to gain full value from ibi products. For a free registration, go to ibi Community.

Legal and Third-Party Notices

Cloud SG software may be available on multiple operating systems. However, not all operating system platforms for a specific software version are released at the same time. See the “readme” file for the availability of a specific version of Cloud SG software on a specific operating system platform.

THIS DOCUMENT IS PROVIDED “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT.

THIS DOCUMENT COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN; THESE CHANGES WILL BE INCORPORATED IN NEW EDITIONS OF THIS DOCUMENT. CLOUD SG MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S), THE PROGRAM(S), AND/OR THE SERVICES DESCRIBED IN THIS DOCUMENT AT ANY TIME WITHOUT NOTICE.

THE CONTENTS OF THIS DOCUMENT MAY BE MODIFIED AND/OR QUALIFIED, DIRECTLY OR INDIRECTLY, BY OTHER DOCUMENTATION WHICH ACCOMPANIES THIS SOFTWARE, INCLUDING BUT NOT LIMITED TO ANY RELEASE NOTES AND "README" FILES.

This and other products of Cloud SG may be covered by registered patents. For details, please refer to the Virtual Patent Marking document located at https://www.cloud.com/legal.