Oracle® Insurance IStream

IStream Publisher Administrator’s Reference Guide

Release 4.2

E14879-01

January 2009

Copyright

Primary Authors: Andrew Brooke and Ken Weinberg

CONTENTS

Chapter 1

Overview

Welcome to the IStream Publisher Administrator’s Reference Guide.

This guide describes how system administrators can administer IStream Publisher and its associated IStream Document Manager components.

This chapter describes:

- *Document Conventions* on page 10
- *IStream Publisher* on page 11
- *IStream Publisher Documentation* on page 12
- *Contacting Oracle for Help* on page 13

Note: For information about how developers and system integrators can integrate IStream Publisher into their systems, see the *IStream Publisher Interface Reference Guide*.

Document Conventions

Tips, Notes, Important Notes and Warnings

`Tip:` A **Tip** provides a better way to use the software.

`Note:` A **Note** contains special information and reminders.

`Important:` An **Important** note contains significant information about the use and understanding of the software.

`Warning:` A **Warning** contains critical information that if ignored, may cause errors or result in the loss of information.

Other Document Conventions

- Microsoft Window names, buttons, tabs and other screen elements are in bold, for example: Click **Next**.
- paths, URLs and code samples are in the Courier font, for example: `C:\Windows`
- values that you need to enter or specify are indicated in the italicized Courier font, for example, *`server_name`*
- values that are optional are indicated with square brackets, for example `[reserved]`

IStream Publisher

IStream Publisher is an enterprise document automation software solution that complements core business systems for product development, sales and marketing, administration and customer service.

IStream Publisher provides a set of integrated services that have been specifically engineered to automate document-intensive business processes. It is used to satisfy event-driven requests, such as new policy fulfillment requests, which trigger a range of activities. These activities can include:

- automatically retrieving documents from multiple sources such as file systems, the web, or IStream Document Manager
- assembling personalized, complex documents and document packages such as policies, letters, contracts and booklets
- rendering them in multiple file formats including: DOC (Microsoft Word), HTML, PDF, PCL, PS (Postscript), RTF, TIFF and TXT; AFP and XML files can be rendered using the Transform service
- delivering them to multiple recipients through their preferred channels (print and mail, fax, email, the web)
- saving them to a file system, FTP or IStream Document Manager

IStream Publisher is available in two editions: IStream Publisher and IStream Publisher *Express*.

IStream Publisher *Expres*s provides a subset of the core Publisher functionality and provides the ability to easily add advanced capabilities as business needs dictate.

IStream Publisher is a J2EE application that uses JMS technology and an XML request-based interface to automate the entire document issuance process from content creation to delivery.

Queues and Requests

You access all Publisher services through queues using either:

- **IBM WebSphere MQ**: IBM’s WebSphere messaging platform
- **OpenJMS**: an open-source messaging platform

The Service Requests are delivered to the various components using JMS text messages (with the body in XML format).

This guide assumes that you have a basic understanding of:

- **IBM WebSphere MQ or OpenJMS**, including the basic messaging functions, and how to set up and manage queues
- the **JMS Message Service**, including architecture and messaging
- **XML**, including an understanding of its structure and styles

IStream Publisher Documentation

IStream Publisher includes the following documents and online help files. If you need a copy of any of these documents, please contact your system or product administrator.

- The *IStream Publisher Release Notes* include general product information, product enhancements and new features, supported platforms and third-party software, assorted considerations, and known issues and limitations.
- The *IStream Publisher Administrator's Reference Guide* helps system administrators configure, control, and manage operations and requests.
- The *IStream Publisher Interface Reference Guide* allows you to integrate IStream Publisher within your own systems. It includes the Software Developers Kit (SDK), which allows you to extend IStream Publisher Publisher, control its operation, and automate requests.
- The *IStream Publisher Schema* is a set of HTML files that describe the structure of Publisher's services and requests.
- The *IStream Publisher Error Messages* contains a list of error messages and their causes.

Contacting Oracle for Help

Customer Support hours are 8:00 AM to 8:00 PM. (Eastern Time), Monday through Friday. Outside of these hours, send us a detailed e-mail message and you will be contacted during regular business hours. Please provide detailed information, as described in the *Support Checklist*.

Contact Information

Mail:	Oracle 145 King Street West, Suite 500 Toronto, ON M5H 1J8
Phone:	1-877-467-9727
Fax:	1-416-642-0315
Email:	markhamsupport-skywire_ww@oracle.com
Web:	www.oracle.com

Support Checklist

When contacting Customer Support, please provide the following information:

- Your name, company name, email address, and phone number
- The exact version number, including any service pack numbers of
  - all your Oracle products
  - your network software
  - your database, including the type
  - Microsoft Windows and Microsoft Word
  - Microsoft .NET Framework
  - your DMS (Document Management System)
- Error messages and the circumstances of their occurrence
- A full description of the problem:
  - What happened? What were the sequence of events that preceded the problem?
  - In which screen or window did the problem occur?
  - Was the problem the result of pressing a key?
  - Did the screen freeze? What functions of the software are affected?
  - How many people are affected?

Chapter 2

IStream Publisher Architecture

This chapter describes:

IStream Publisher and Its Architecture

IStream Publisher is a scalable document fulfillment solution that produces highly personalized documents in client-preferred formats and delivers them through multiple channels.

IStream Publisher receives messages (or “requests”) from queues, interprets the XML request in the message, and then executes the instructions in the message.

Scalability

IStream Publisher has been designed as a distributed system where the workload is balanced across many Worker computers. These Workers all have the same structure, in the sense that they all run one or many Service Manager components. Service managers constitute the controlled runtime environment where services are deployed and executed. Vertical scalability is therefore attained by increasing the number of Service Managers or listeners running on a Worker while horizontal scalability relies on an increasing number of Worker computers.

Reliability and Fault Tolerance

IStream Publisher makes extensive use of Message Queues to provide a highly reliable and fault tolerant system. At the same time IStream Publisher allows for a high level of component redundancy, which can easily increase its availability.

Flexibility

With its functionality delivered as services, IStream Publisher can be easily extended to support new functionality without changing its architecture. The Software Development Kit (SDK) provides users with the ability to create their own services.

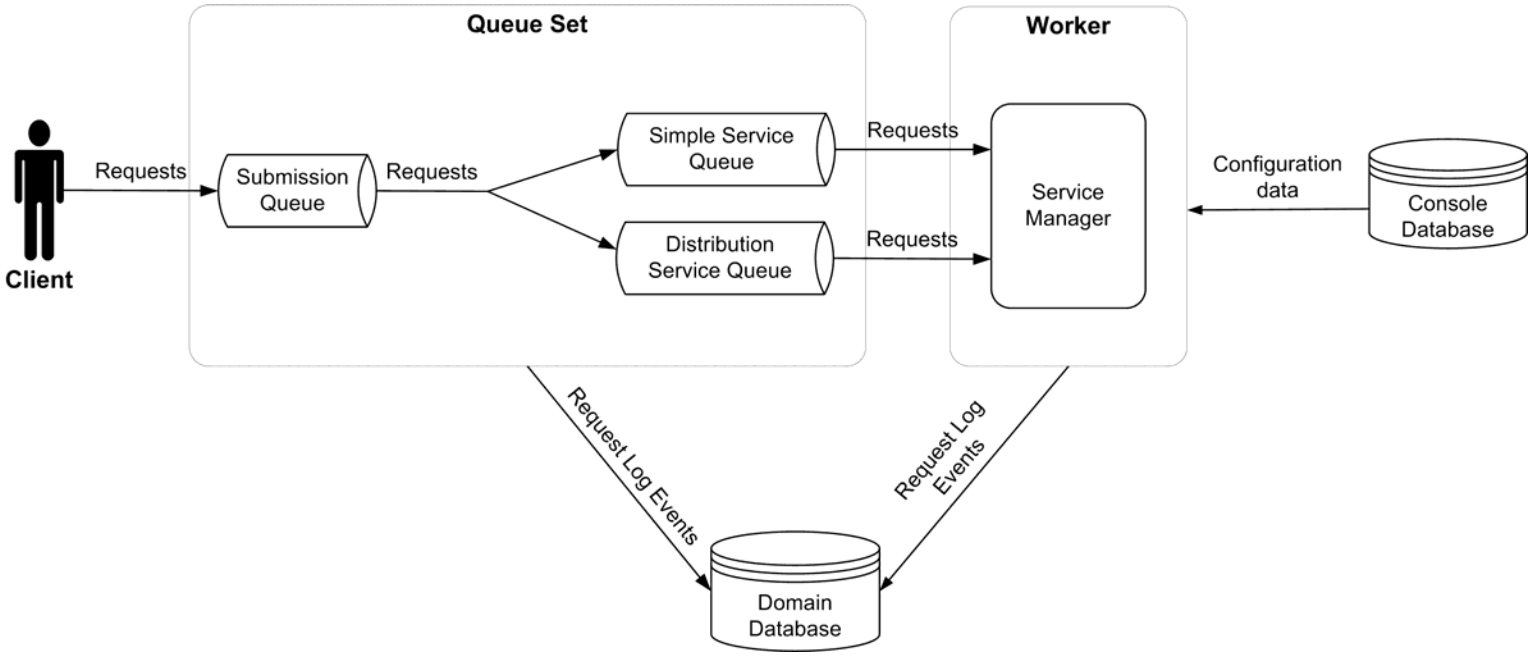
The following sections present IStream Publisher's architecture from a functional as well as an administration perspective.

IStream Publisher's Design and Process

In IStream Publisher:

- All components receive their configuration information directly from the **Console Database** when the system starts.
- The system uses **Queue Sets**, which are a logical grouping of queues based on the types of requests in the queues, in order to create a logical unit of work. Examples of **Queue Sets** include grouping by line of business, department, location, application, and document type. At least one queue set is required.
- A **Queue Set** includes its name, and the names of the **Submission Queue**, the **Simple Service Queue**, and the **Distribution Service Queue**.
- A **Queue Set Reference** (**QueueSetRef**) is assigned to each Worker. It contains the number of listeners and the name of the **Queue Set**.
- All **Request Log Events** are recorded directly into the **Domain Database**.
- All components retrieve their configuration information directly from the **Console Database.**
- All agents are part of the **Worker**.

The following diagram gives an overview the process:



Functional Components

The basic functional unit in IStream Publisher's architecture is the Service Manager where the services that deliver the functionality run. A series of other components support the coordination of multiple Service Managers and help trace request processing.

The main functional components are:

Service Managers

These are deployed on Worker computers and serve as the environment where services run. Service Managers are responsible for fetching requests from Distribution Queues and dispatching them to the appropriate service.

Services

These are the components responsible for doing the actual work required to fulfill requests issued by client systems. A Simple Service performs a very specific operation like generating a document, rendering from one format to another or delivering content through a particular Delivery Channel. Also there can be composite services that rely on Simple Services to perform a more complex operation.

The service managers run in the Windows Worker service.

Queue Sets

There are a series of queues used by IStream Publisher for different purposes, such as interfacing with client systems and supporting communication among the IStream Publisher components.

Submission and Service Queues

The Submission Queue is used to interface with the client system. The Worker retrieves messages from Submission Queue, which moves the request messages into the Service Queue.

Response Queue

This is the queue where Service Managers place the responses for the processed requests. This queue is optional.

Completion Queue

When requests are successfully completed, a completion message is sent to the Completion Queue. It is used internally by the Distribution service in the processing of a Distribution Request to sequence its spawned Simple Requests.

Client

This is the system that makes use of the functionality offered by IStream Publisher. Client systems interact with IStream Publisher via Message Queues. They place requests in Submission Queues and receive responses in Response Queues. IStream Publisher also provides a Java API and Web Services interface that allows a client system to interact with IStream Publisher programmatically.

Administration

IStream Publisher administration architecture relies on the worker. The functioning of the system can be controlled via administrative operations that are processed by the workers.

The following components support the management features of IStream Publisher:

IStream Publisher Console

A web interface that allows the administrative user to manage the system. The IStream Publisher Console allows the administrative user to invoke the administrative operations exposed by the components in the system. It also allows the administrative user to browse and change the static configuration of the system.

Requirements and Servers

IStream Publisher Console is a servlet-based application and therefore needs a Servlet Container. In addition to Apache Tomcat, IStream Publisher Console can be deployed on any supported application servers. For a list of supported application servers, please see the *IStream Publisher Release Notes*.

A database is required to store various information about the system, such as users in the system, static configuration of the components, and so on.

Worker

The Worker:

- controls the processes where the Service Managers run
- provides information about the Worker
- can only service one Worker machine
- handles global system administrative operations by coordinating the work of the other agents.
- manage the queues and the requests they contain.
- allows an administrator to do lookups for requests in the system and provides information with regard to their status and metadata

Simple Requests

Simple Requests are the basic means by which IStream Publisher functionality can be invoked by client applications. They are useful when processing one particular task such as document generation, format rendering and content delivery. Service Requests are picked up from Service Queues and directly processed by Service Managers.

The limitation of a simple request is that it is not capable of sending more than one task within one XML request.

When forming Simple Requests, client applications are directly responsible for managing the input and the output files required by the Service. The location of these files is typically specified by their URLs, with some exceptions, when UISRs are still used to indicate model documents.

Aggregate Requests

IStream Publisher uses the Aggregate Request type as a simple way to enforce the sequencing of Simple Requests. The Aggregate Request is a hierarchical packaging of Simple Requests into a tree of requests.

For example, if you have to Generate a Document, render it from one format to another and finally deliver the document to a recipient, this type of request is ideal because it uses dependencies to complete one task before moving on to the next.

Aggregate Requests are processed in a recursive manner. The Service Manager receiving an Aggregate Request identifies the root Simple Request and processes it. It then takes all the subrequests and resubmits them as Aggregate Requests. This process continues with the next Service Managers that pick up sub-aggregates for processing. When there are no more subrequests, the processing ends.

Limitations

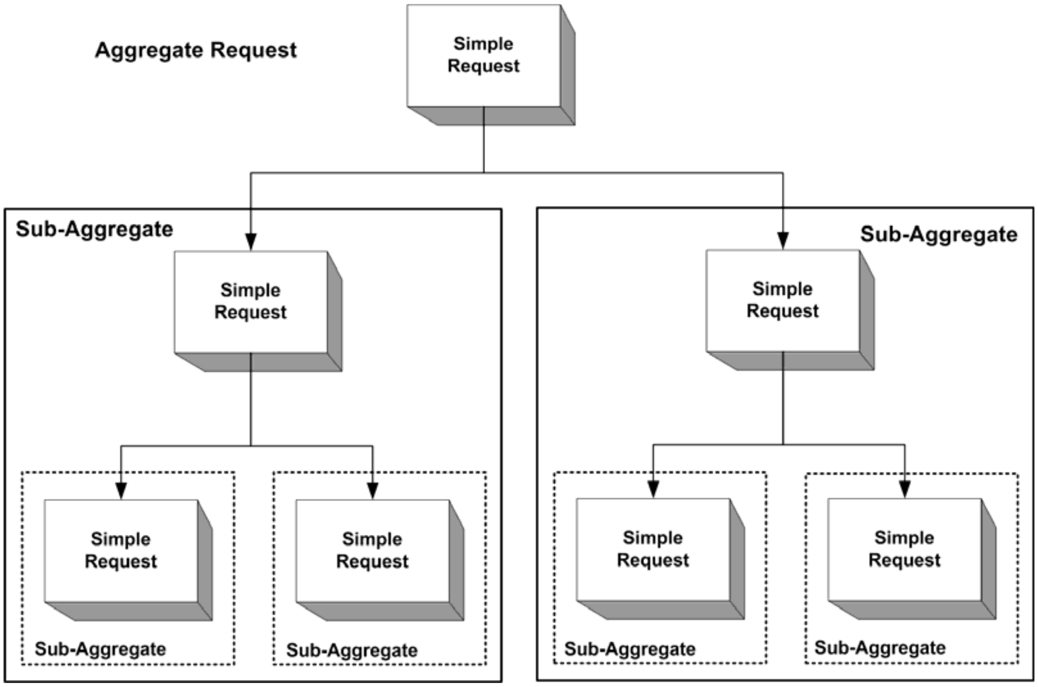
One limitation of the Aggregate Request is that there is no consolidated response message for the Aggregate request telling you that all simple requests have completed. Instead, there are multiple responses to provide the information that all the requests contained in the Aggregate Request have completed. The only way to get this information, in the context of an Aggregate Request, is to:

- know in advance how many Simple Requests were contained in the Aggregate Request, and
- check the Request Log to make sure all the requests have completed

Another limitation is that the issuer of the request must correlate the output from a service with the input for the next by placing files in the appropriate shared locations.

Both of these limiting characteristics and others are addressed by the Distribution Service.

Aggregate Request Diagram



Distribution Requests

Distribution requests are used to invoke a Document Distribution operation. Document Distribution, as defined by IStream Publisher, consists of the distribution of a package of interrelated documents to a list of recipients, in various formats and through different Delivery Channels.

Efficiencies Gained

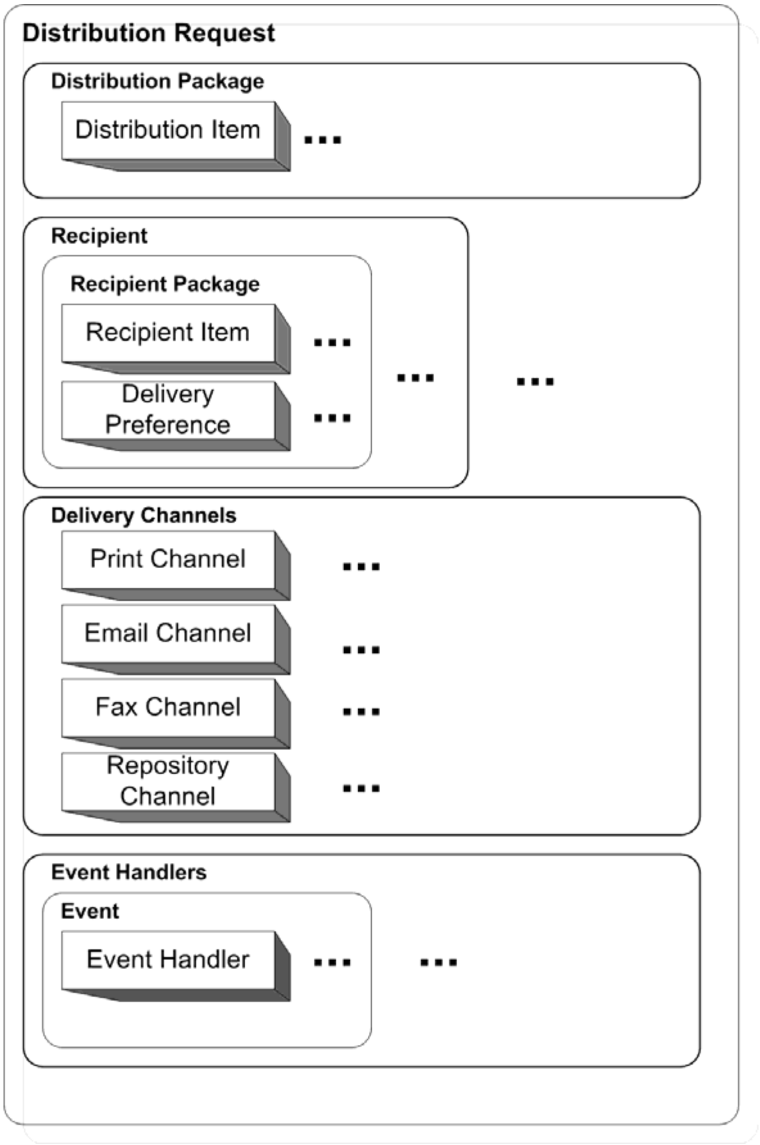
A Distribution Request is processed by resolving it to the set of Simple Requests required to fulfill it and then submitting them for processing in the appropriate order. The processing of a Distribution Request takes advantage of the fact that the same set (or sub-set) of documents is distributed to many users and thus reduces the number of document generation and render operations.

Intermediate Files

IStream Publisher manages all of the intermediate files that are created. It ensures that files produced by a service are available for the other services that depend on it. At the end of the processing it cleans up (removes) all unnecessary files.

Each of the recipients receives either the whole Distribution Package or a subset of it. A package received by an individual recipient through any number of Delivery Channels is called a Recipient Package. A recipient can receive any number of Recipient Packages.

The Distribution Request



Event Handlers

The processing of a Distribution Request can be customized via Event Handlers. As the processing of the request progresses, various checkpoints are reached at which time events are raised. The issuer of the request can specify in the request what services it wishes to be invoked when particular events are raised.

Declarative Format

In contrast to the Aggregate Request where the raw Simple Requests are listed in a hierarchical order, the Distribution Request has a declarative format where the issuer specifies only:

- the contents of the Distribution Package

- the recipient(s) with their packages
- the Delivery Channels
- the potential Event Handlers

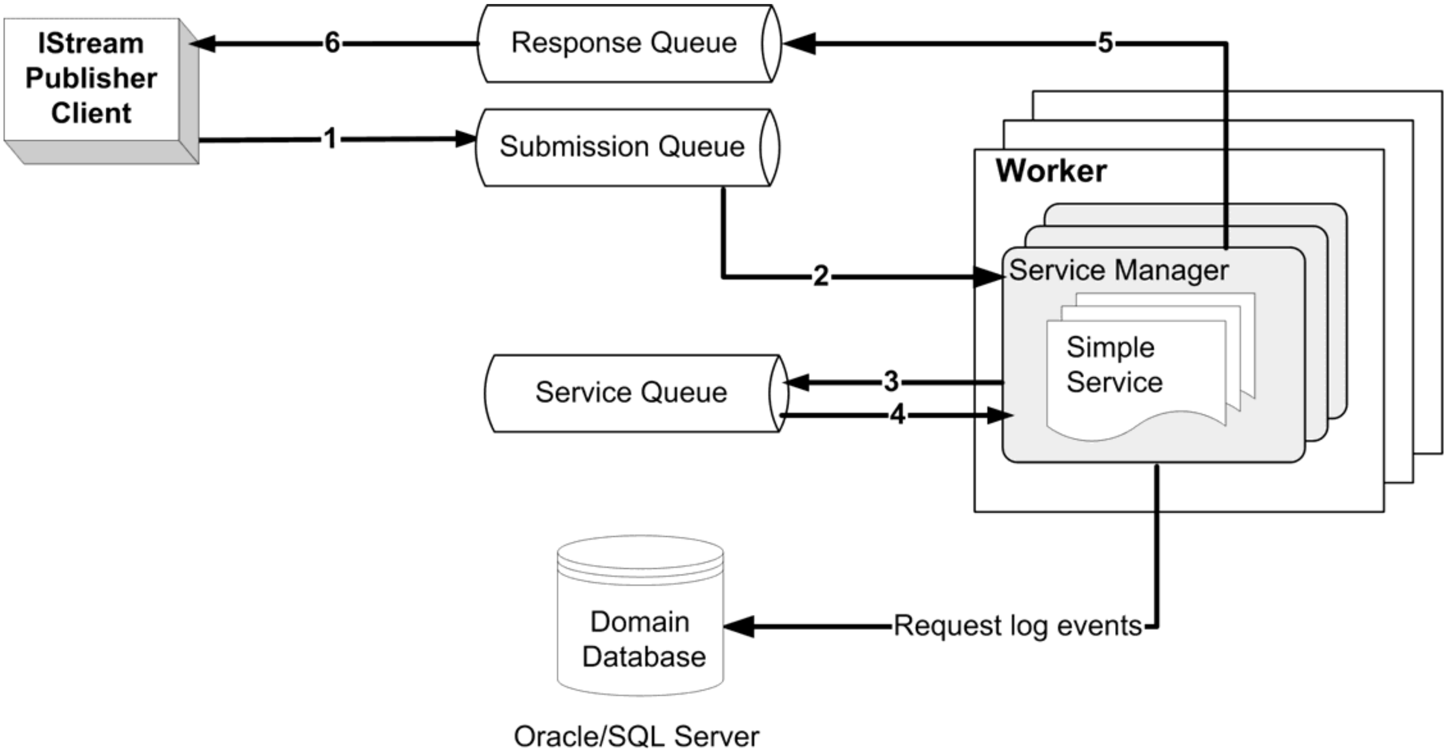
The Distribution Service interprets the request. It determines what Simple Request(s) it must issue in order to satisfy the Distribution Request, as well as the order of those Simple Requests.

Note: By using a declarative format for Distribution Requests, IStream Publisher insulates the business application requiring the Document Distribution functionality. As a result, the user or client application needs no prerequisite knowledge with regard to how the distribution is performed.

The Distribution Request diagram (on page 25) presents the main elements of the Distribution Request. The ellipsis indicates that the preceding element can occur many times. For example, the recipient element is followed by an ellipsis since there could be many recipients specified in a Distribution Request.

The Simple Service Request Process Flow

The processing of a Simple Service request involves a series of steps and uses most of IStream Publisher's functional components, as the following diagram illustrates:



The steps are as follows:

1. The **IStream Publisher Client** submits a request to the **Submission Queue**.
2. The **Service Manager** retrieves the request from the **Submission Queue** and changes the request's status to *Pending*.
3. The **Service Manager** moves the request to the **Service Queue**.
4. While the previous steps are occurring, the **Service Manager** retrieves the request from the **Service Queue** and processes it.
5. The **Service Manager** sends a message to the **Response Queue** indicating that the request processing has been completed.
6. The Client receives a response status from the Response Queue.

During this process, **request log events** are logged to the **Domain Database**.

If the client application does not specify a Response Queue in the request's headers, steps 5 and 6 are skipped. Also, depending on the configured request error logging level (see *Error Log Levels on page 89* of the *Interface Reference Guide*), some or all of the request status update messages might be skipped.

The Distribution Service Process Flow

The Distribution Service process flow contains all of the steps in the Simple Service process flow, but also includes the following *additional* steps. (These steps replace steps 4 and 5 of the Simple Service process flow.)

4 – The **Service Manager** retrieves the request from the **Service Queue**.

4.1 – The **Distribution Service** analyzes the distribution request. The request is broken down into several simple requests.

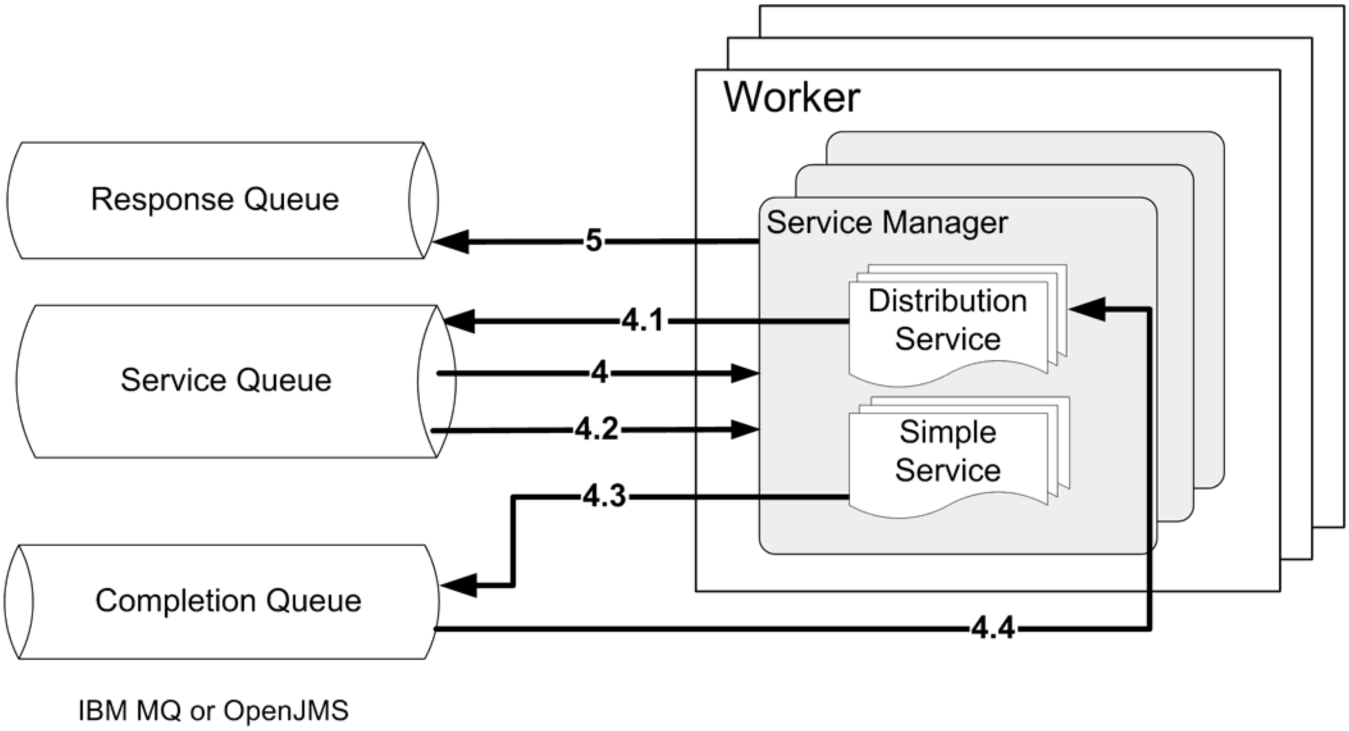
4.2 – The **Service Manager** receives the simple request from the **Service Queue**.

4.3 – The **Service Manager** places a message in the **Completion Queue** indicating to the **Distribution Service** that the processing of the request is complete.

4.4 – The **Distribution Service**, which was waiting on the **Completion Queue**, gets the completion message from the **Completion Queue**. The **Service Manager** is made aware that the **Distribution Request** has successful completed.

5 – The **Service Manager** sends a message to the **Response Queue** indicating that the request processing has been completed.

The following diagram indicates these extra steps:



The Distribution Request Lifecycle

There are two important stages in the processing of a Distribution Request. In the first stage the Distribution Request is interpreted and a Task Graph is created. This Task Graph reflects all the Simple Services and Event Handlers that must invoked in order to process the Distribution Request. The Task Graph also reflects the dependencies that exist among the Simple Service invocations.

Distribution Request Processing Steps

The submission of the Distribution Request to a Submission Queue by a client application and the receiving of the Response are the same as those for a Simple Request. In between these events, the individual stages involved in the processing of a Distribution Request are:

First Stage

The Task Graph together with the information in the Distribution Request are persisted into a database as the Distribution State. This is required because the Distribution Request is processed in a distributed fashion and all the Workers process the request must access this state in a shared location.

Second Stage

This stage consists of iteratively going through the task graph and determining the next simple service(s) to be performed. Each simple service is submitted to the Distribution Queue defined in the Distribution Service configuration. The Distribution Service receives responses for simple services involved in executing a particular Distribution Request on a Completion Queue.

When there are no more simple services to execute and all the responses are received from the Completion Queue, the Distribution Service return a response to Service Manager.

Failover

Failover refers to the ability of IStream Publisher to guarantee the processing of Service Requests even if components fail and must be restarted.

IStream Publisher has the following failover capabilities:

- Dynamically and transparently cope with component failure.
- Restart and recover in the event of a complete system failure.

Queue Failover

Queue failover allows another queue to take over processing if a queue fails. JMS provides a failover architecture through the clustering of the queue and replication of the message (request) repository.

Worker Failover

Worker failover allows a request to be sent to another Worker for processing if any Worker fails. In order to accommodate this, the IStream Publisher architecture has been designed as follows:

- Multiple Service Managers running on different Workers can be listening on the same Service Queue. If one Worker fails, other Workers can be available to process the request.
- Whenever a Service Manager receives a request from a queue, it does not acknowledge the receipt to the queue until the request has been processed.
- As a a result of this, if the Service Manager fails before the request is completed, the acknowledgment is not sent and according to the operation of JMS queues, the request is re-queued and then subsequently is taken by a Service Manager on another Worker Machine.

Restart and Recovery

IStream Publisher has the ability to continue processing requests where it left off after being restarted due to a failure in the system.

Queues

Recovery is supported by the JMS persistent mode. Persistent mode indicates that the content and state of the queue is continuously saved to a persistent repository (for example, a file system) rather than just remaining in memory.

This allows it to be recovered if the system fails. The queue stores enough information to guarantee that each request is delivered and that it is not duplicated.

Worker

Recovery on a Worker machine is straightforward since the Worker contains no states that must be persisted. If a Worker fails, the request is re-queued.

Chapter 3

IStream Publisher Administration

This chapter describes the Administration Model of a set of managed components that perform administration operations, contain administration information and raise administrative events.

It describes the more general aspects of the Administration Model, including the names of the managed components, some of the events that the components raise, component startup, shutdown and state management. Commands related to system startup and shutdown are also described. These commands relate to the system as a whole.

This chapter describes:

- *The IStream Publisher Administration Model* on page 32
- *Deferring Requests* on page 35
- *The Request Log* on page 39

The IStream Publisher Administration Model

This section describes:

- *Managed Component Names* on page 32
- *Attributes and Commands* on page 33
- *State Management* on page 33
- *System State and Configuration* on page 34

Managed Component Names

Each managed component is identified in the system by a unique name. The name is a JMX (Java Management Extension) object name and has the following syntax:

```
domainName:property=value[,property=value]*
```

`domainName` is a case-sensitive string that provides a naming space for a particular IStream Publisher deployment.

The property-value pairs that make up the name can occur in any order. The following property names are standard:

- **type** – One of the managed types. This is a mandatory property for the Service Manager and Service only.
- **name** – A case-sensitive string assigned by the user to identify the managed component.
- **host** – The host name of the computer where the component runs. Note that host names cannot begin with a number.

For example, the name of a managed component representing an IStream Publisher Service Manager could be:

```
ABCompany:type=ServiceManager,host=server1
```

Attributes and Commands

In addition to the component name, managed components support many different types of attributes, particularly those related to the component type. For example, a component of the Service Manager type has an attribute for the number of listeners.

State Management

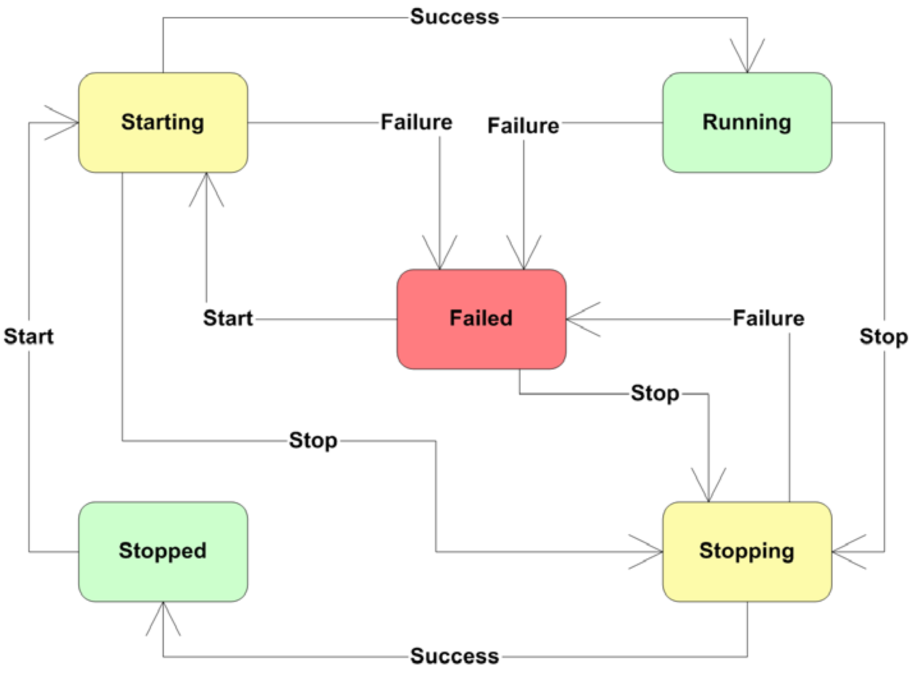
Components implement a state model. Their state can be controlled and queried through admin commands.

When a component is instantiated (starts up), it implicitly goes into the STOPPED state. It can also go into the FAILED state in the case of an abnormal startup. A component can be shut down from any state.

While being instantiated, a component can go through different states with regard to its functional duties, or the actual work it performs. A component can be idle and not accept any requests to perform work, and is therefore in the STOPPED state. Alternately, it can be active and perform its duties, and is therefore in the RUNNING state. There are also transient states (STARTING and STOPPING) and an error state called FAILED.

The following diagram illustrates the possible states of a component, and the progression from one state to another:

Progression Between States for an IStream Publisher Component



The states of a managed component have the following meanings:

STARTING – This state indicates that the component has been requested to start and is in the process of starting.

RUNNING – This is the normal running state for a component. This state indicates that the component is operational.

STOPPING – This state indicates that the component has been requested to stop, and is in the process of stopping.

STOPPED – This state indicates that the component has stopped and can be restarted.

FAILED – This state indicates that the component has unexpectedly stopped. It is possible to start a component that has entered the FAILED state by calling the start operation. It is also possible to stop a component that has entered the FAILED state by calling the stop operation.

Admin commands can be used to control what state a component is in.

This state diagram also shows when it is appropriate to invoke the start and stop commands.

Start and Stop Commands

Starting and stopping applies to components that perform a functional role. These components are:

- **Service Manager** – When running, it picks up requests from the Service and Loopback Queues and dispatches them to services.
- **Service** – When running, it processes requests.

Workers have a strictly administrative role and therefore they do not need to be started or stopped. At any point in time, the Workers can only be in one of two states: RUNNING and FAILED

Note: As mentioned above, there are no start or stop commands for these components. Therefore, once one of these components reaches the FAILED state, the operating system service or daemon process where it runs will have to be restarted in order to bring it back into the RUNNING state.

System State and Configuration

The Worker maintains the state and the configuration of the system. The system configuration consists of the topology of the system (that is, the hierarchical relationships between the various types of components) and their managed attributes. For example, for Service Manager components, the number of listeners is a managed attribute.

The Worker also keeps track of the runtime state of all the individual components in the system.

When running in failover mode with more than one Worker, the system state and configuration are shared among the multiple instances of the Worker.

The system state can be obtained with the **QueryState** command.

Deferring Requests

IStream Publisher provides the ability to prepare documents now, and deliver them later. This ability is called Deferred Request processing, and it includes two somewhat different meanings: delayed request processing as well as scheduled request processing.

IStream Publisher accepts deferred delivery dates in all types of requests: Simple, Aggregate, or Distribution.

Benefits

There are many uses and benefits of this functionality:

- You can delay the processing of delivery requests until a point in time when there is less traffic in the system. If you want to run a lengthy business procedure, you can specify the request to run at a specific scheduled date and time. This will allow the utilization of your network infrastructure and resources in the most effective manner.
- You can do the initial part of a request but delay the delivery part until a later time, when it can take place unattended.
- You can generate and render a high volume of documents during your business hours to make sure that the generation and rendering are both error free. Meanwhile, the delivery can take place during non-business or off-peak hours without further human involvement.
- The delivery can be scheduled to run at night, when your telephone lines tend to have lower traffic and the rates are discounted.

Deferred Request processing means deferring the start of processing a given request in the system but does not guarantee either the certain start time of processing or the time of the physical delivery. A Deferred Request will be processed not sooner than the specified deferral time and not later than the expiration time.

Important: The time on all IStream Publisher machines must be synchronized.

General Requirements for Deferred Requests

- A request can be submitted in deferred processing mode or can be deferred by Admin Commands while it is in a Service Queue.
- Messages with a deferred delivery date are held in the system until the deferred delivery date is reached.
- The system validates conditions and controls execution of all Deferred Requests by making requests available for processing on the time specified in the deferred condition.
- To get an update on the status of a deferred request, the administrator can directly query the system through admin commands.

- The request will be rescheduled from the point in time at which the request was initially submitted to the system for processing.
- Requests that cannot be processed within the expiration interval are automatically removed from the queue and logged in the Request Log as cancelled. Expired requests can easily be resubmitted from the Request Log. The user must decide on a case-by-case basis how to resubmit the request.
- Using admin commands, the user can either change the deferred setting or resume (one or many) deferred request(s) using the selector parameter, and process it now.

Note: The system logs the deferral and expiration time of all requests in the Request Log.

Important: If the **DeferralTime** or **ExpirationTime** field is in an invalid format, this option will be ignored and the request will be processed as a regular request.

Deferring Requests in Different Time Zones

If a client machine that submits requests to IStream Publisher and the IStream Publisher system are in different time zones, the Deferred Request is processed according to the IStream Publisher system time zone.

Submitting a Deferred Request to IStream Publisher

The system can recognize submitted requests in the deferred state for further processing and process them according to the deferred date and time.

The client application uses the **DeferralTime** and **ExpirationTime** JMS header message fields with a predefined format to schedule Deferred Request processing.

The valid format for the **DeferralTime** and **ExpirationTime** JMS header fields are as follows:

Parameter	Attribute	Description
DeferralTime	Fixed Time	**Dmm/dd/yyyy, Thh:mm**, where: • D - mandatory letter for fixed date • mm/dd/yyyy- date format; • T - mandatory letter for fixed time; • hh:mm - time format, where hh is in 24 hours format; • D/05 - a Request is processed on 5th day of current month, year • D1//2003- a Request is processed on January 2003. • D01/02/2002 - a Request is processed on January 2, 2002; • D/15/ - a Request is processed on 15th day of current month, year • D//2003- a Request is processed in 2003 year.
	Timer	+Dmm/dd, Thh:mm, where: • D - mandatory letter for date parameters in timer • mm/dd- number of months, days; • T - mandatory letter for time parameters; • hh:mm - time format, where hh is in 24 hours format; • +D/05 - a Request is processed in 5 calendar days • +D01/- a Request is processed in 1 month. • +D01/02 - a Request is processed in 1 month and 2 calendar days.
ExpirationTime	Fixed Time	**Dmm/dd/yyyy, Thh:mm, where:** • D - mandatory letter for fixed date • mm/dd/yyyy- date format; • T - mandatory letter for fixed time; • hh:mm - time format, where hh is in 24 hours format •
	Timer	**+Dmm/dd, Thh:mm, where:** • D - mandatory letter for date parameters of timer • mm/dd- date format; • T - mandatory letter for time parameters in timer; • hh:mm - time format, where hh is in 24 hours format;

The administrator can defer and resume Deferred Request(s) at any time using the following admin commands:

DeferRequest

Use this command to defer a request (Simple, Aggregate or Distribution) that is already in a queue, or to reschedule an already deferred request, if the request has not been processed.

ResumeDeferredRequest

Use this command to resume one or more previously Deferred Requests, releasing them for final distribution.

No “Start Time” Guarantees

Deferred Requests are processed according to the specified date and time according to the system's availability. There is no guarantee with regard to the start time for the processing of potentially high-volume requests, which may have the same deferred parameters. Deferred tasks are executed in sequence, depending on the configuration of the system as well as the number of listeners.

The processing occurs after a request has been resumed. When a request is resumed, the system must:

- Resume the deferred request
- Update the status in the Request Log, and then
- Start processing the request.

Because Deferred Request processing can happen relatively slowly, enough time can elapse between scheduling and execution. A Deferred Request attribute allows or restricts request processing after the expired time. An administrator can resubmit the expired requests from the Request Log. The administrator can choose to reschedule Request(s) by using the admin command.

The Request Log

The Request Log component assists in tracing the location and status of all requests in the systems. The possible status values logged for a request are:

- Pending
- Paused
- Processing
- Completed-Success
- Completed-Success-With-Info
- Completed-Failure
- Completed-Canceled
- Resubmitted
- Deferred

To perform administration on a request, the **findRequests** command must first be used to identify a request using metadata (for example, the submitter name). With the request ID obtained this way, the **RequestInfo** can be used to return detailed information about the request.

Each request spawned from an aggregate or composite request has a main request ID and an aggregate or composite request ID in addition to its own request ID.

When tracing an aggregate or composite request there are multiple responses for each Simple Request.

Chapter 4

The IStream Publisher Console

This chapter provides an overview of the IStream Publisher Console. System administrators can use the IStream Publisher Console to configure and manage IStream Publisher.

This chapter describes:

- *The IStream Publisher Console* on page 42
- *Using the IStream Publisher Console* on page 43
- *General Links* on page 45
- *Changing Your Password* on page 46

The IStream Publisher Console

The IStream Publisher console allows system administrators to configure and manage IStream Publisher. Using the IStream Publisher console, you can:

- perform run-time administrative operations
- define configuration settings
- monitor and manage several IStream Publisher installations

All components that can be managed from the IStream Publisher console store their configuration settings in the IStream Publisher *console database*.

The IStream Publisher console database contains the following data:

- **run-time data** – data used in the normal operation of IStream Publisher, such as the object model instance data
- **configuration data** – data created or modified during authoring, customization or personalization of a component.

For detailed information about the IStream Publisher console database, see *Configuring Entities & Attributes in the Console* on page 53.

Note: The IStream Publisher console supports *roles* to control each user's access to certain functions. The various procedures describe which roles can perform each function.

Using the IStream Publisher Console

This section describes how to use the IStream Publisher console, and explains how to navigate through the program.

This section describes:

- *Logging into IStream Publisher Console* on page 43
- *Navigating the IStream Publisher Console* on page 43

Logging into IStream Publisher Console

To log in to the IStream Publisher console, you need a user ID, password and at least one role. Roles control the access to IStream Publisher console modules. For more information about roles, see *IStream Publisher Roles* on page 48.

Note: To allow a systems administrator access to the IStream Publisher console before users have been set up, it is installed with a single predefined Administrator user.

Method: Log into the IStream Publisher console

1. Open the **IStream Publisher Console Login** page:

   `http://`*`server_name`*`/rmc`

   where *`server_name`* is the server where you installed **IStream Publisher**. This may include a port number.
2. Enter your **User ID** and **Password**. Note that the password is case-sensitive. If you do not have this information, contact *Customer Support*.
3. Click **Log In**.
4. The **IStream Publisher Console** opens.

   If you are logging on for the first time with your own user ID and password, you must change your password immediately after logging in. The **Change Password** page opens automatically.

Navigating the IStream Publisher Console

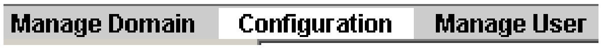
Every page in the IStream Publisher console contains a **Module Menu** and **General Links**.

Warning: Do not use the forward or back buttons of your web browser when a folder is selected in the tree structure, because this may cause errors. If you inadvertently use these buttons, click **OK** on the message and then click the folder at the top of the tree structure to continue.

Module Menu

The *Module menu* is a gray bar near the top of your screen. It indicates which module you are using, and allows you easily go to another module by clicking its name.

The module menu appears as:



The three main modules are:

- **Manage User**: see *Managing Users in the Console* on page 47
- **Configuration**: see *Configuring Entities & Attributes in the Console* on page 53
- **Manage Domain:** see *Managing Domains in the Console* on page 63

Note: Only users with the Administrator role will see the **Manage User** tab.

If you choose a new module from the menu and you have not saved your work, you are given the option to remain in the module so that you can save your work, or exit the current module without saving.

General Links

These links are displayed in the upper right corner:

Help | Change Password | Logout

With the exception of **Logout**, these links do not affect any task you are performing:

- **Help** – opens the online help
- **Change Password** – allows you to change your IStream Publisher password
- **Logout** – logs you out of IStream Publisher

Changing Your Password

After logging in to IStream Publisher, you can change your password according to the rules set by your IStream Publisher Administrator. (See *Password Guidelines* on page 46.)

Method: Change your password

1. Click the **Change Password** link in upper-right corner of the screen.

2. The **Change My Password** page opens. Your user ID is automatically entered into the **User ID** field.

3. Enter your existing password into the **Current Password** field.

4. Enter a new password into the **New Password** field. Your new password cannot be the same as your existing password.

5. Enter the new password again into the **Confirm New Password** field.

6. Click **Submit**.

Your password is changed.

Password Guidelines

Your IStream Publisher administrator or system administrator can give you the guidelines to follow when creating your password, including:

- its minimum and maximum length
- whether it requires numbers or special characters
- whether it requires both upper and lower case letters

If you are having problems creating passwords, contact your system administrator to see if these requirments have changed.

Note: All passwords are case-sensitive.

For security reasons, ensure that your password:

- is different than your user ID
- does not contain personal details such as your family name, names of relatives or date of birth
- is easy for you to remember but hard for others to guess

Important: You should memorize your password. Do not write it down, and do not share it with others.

Chapter 5

Managing Users in the Console

Administrators can use the **Manage Users** module to create, view, edit and delete IStream Publisher console users.

This chapter describes:

- *IStream Publisher Roles* on page 48
- *Creating a New User* on page 49
- *Viewing or Editing a User* on page 50
- *Deleting a User* on page 51

IStream Publisher Roles

The IStream Publisher console uses roles to control which modules a user can access. Each user must be assigned at least one role.

Administrators can assign roles when creating a new user or editing an existing one: see *Creating a New User* on page 49 and *Viewing or Editing a User* on page 50.

The IStream Publisher console roles are:

- **Administrator** – users with this role have access to all modules in the IStream Publisher console: **Manage Domain**, **Configuration** and **Manage User** modules
- **Operator** – users with this role have access to the **Manage Domain** and **Configuration** modules only

All the procedures throughout this guide indicate which roles can perform the described function.

Using the Predefined Administrator User

A predefined Administrator user is included with IStream Publisher console. This user has full access to the application, therefore access to this user ID should be restricted.

A trusted system administrator should use this user ID only to log into the IStream Publisher console the first time, access the **Manage Users** module, and set up user accounts for all the administrators who must use the IStream Publisher console (including an account for themselves). Afterwards, all users should always log in using their own user ID.

Creating a New User

Only users with an **Administrator** role can create new users.

Note: When an Administrator creates a new user, they need to enter the user's ID and initial, temporary password. For security reasons, as soon as the new user logs in the first time, the user must change their password.

Method: Create a new user

1. On the modules menu, click **Manage Users**.

The **Manage Users** page opens.

2. Select the **Create a New User** radio button.

3. Click **Submit**.

The **Set Up User** page opens. Note that all fields are mandatory.

4. Enter the user's ID into the **User ID** field. Each user must have a unique ID.

5. Enter the user's information into the **First Name**, **Last Name**, and **Email Address** fields.

6. To activate the user and give them immediate access to the IStream Publisher console, select **Yes** in the **Profile Enabled** section. Select **No** to keep the user's profile but not allow them to log in to the IStream Publisher console.

7. Enter the user's password into the **Password** field.

8. Enter the password again into the **Confirm Password** field.

9. Select a role in the **List of Roles** list on the left side.

10. Click **>** to move a selected role into the **Assigned Roles** list and assign it to the user.

To remove any role from the **Assigned Roles** list, select the role, then click <.

11. To save your changes, click **Save & Exit**.

The new user is saved.

Once you have created and assigned roles to the user, you can review the user account, and make any necessary changes: see *Viewing or Editing a User* on page 50. Once you have finalized the information in the user's account, the user can use their user ID and password to log into the IStream Publisher console.

Viewing or Editing a User

Only users with an **Administrator** role can view or edit a user's record. They can edit a user's name, email address or password, enable or disable a user, or change the roles assigned to them.

Note: An Administrator cannot change the **User ID** of an existing user. Instead, the Administrator must delete the user (see *Deleting a User* on page 51), and then create a new user (see *Creating a New User* on page 49).

Method: View or edit a user

1. On the modules menu, click **Manage Users**.

The **Manage Users** page opens.

2. Click **Open a user to view or edit**.

A drop-down list of the current user names appears.

3. Click the down arrow in the drop-down list to select a **User ID**. (Users are listed in order of their **User ID**.)

4. Click **Submit**.

The **Edit User** page appears, displaying the user's profile information. Note that you cannot change the **User ID**.

5. Make any of the following changes:

- change the user's first name, last name or email address
- allow or prevent the user from being able to log into IStream Publisher console by selecting **Yes** or **No** in the **Profile Enabled** section
- change the user's password by entering a new password in the **New Password** and **Confirm New Password** fields

6. To modify the roles assigned to the user:

a. Select a role in the **List of Roles** list and click > to move it into the **Assigned Roles** list.

b. Remove any role from the **Assigned Roles** list by selecting it and clicking <.

c. Use the >> and << buttons to move all roles from one list to the other.

7. Click **Save & Exit.**

The revised user is saved.

Deleting a User

Only users with an **Administrator** role can delete a user.

Note: You cannot delete the predefined Administrator user that is installed with the IStream Publisher console.

Method: Delete a user

1. On the modules menu, click **Manage Users**.

   The **Manage Users** page opens.

2. Select **Remove a User.**

   A drop-down list of the current user names appears.

3. Click the down arrow in the drop-down list to select the user you want to delete. (Users are listed in order of their **User ID**.)
4. Click **Submit**.
5. You are prompted to confirm that you want to remove the user.
6. Click **OK** to remove the user or click **Cancel** to return to the **Manage Users** page.

Chapter 6

Configuring Entities & Attributes in the Console

This chapter describes:

The IStream Publisher Console Database

The IStream Publisher console is an interface to a database that acts as a central container for all the IStream Publisher component elements. (Components include the various IStream Publisher agents and services, but are not actually part of the IStream Publisher console database.)

Hierarchy of the IStream Publisher Console Database

The IStream Publisher console database stores the configuration values for all the components that the IStream Publisher console controls. It has a hierarchical structure that contains the following elements:

Attributes and Values

In the IStream Publisher console database, data is stored as *attributes* and their values.

Note that:

- You cannot create, delete or rename attributes; you can only change the values assigned to attributes.
- Some attributes are *required*: these must have a value assigned to them.

Entities and Entity Instances

Attributes are grouped into *entities*.

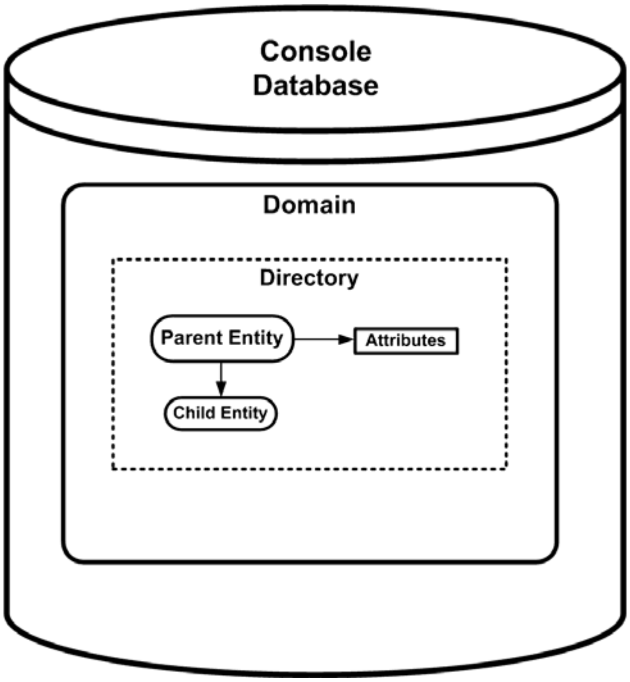
Note that:

- An entity can contain other entities (*a parent entity* contains a *child entity).*
- You can create copies of entity types, called *entity instances*: see *Creating a New Entity Instance* on page 59.
- You cannot add, edit or delete entities, only entity instances.

Directories and Domains

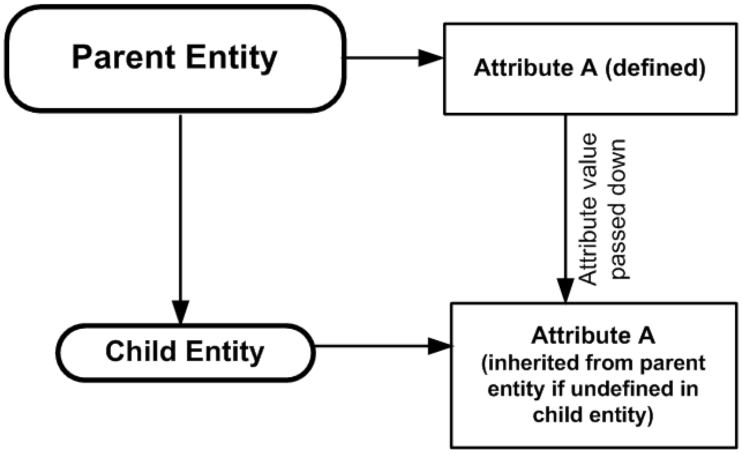
Entities are grouped into *directories*, which in turn are grouped into *domains.* Finally, domains make up the IStream Publisher console database.

The following diagram illustrates the relationship between these items:



Inheriting Attribute Values

If an attribute in a child entity is undefined, the attribute's value can be passed down or "inherited" from its parent entity, as illustrated in the following diagram:


Parent Entity
Attribute A (defined)
Attribute value passed down
Child Entity
Attribute A (inherited from parent entity if undefined in child entity)

If no value is found in any of the parent entities, the default value is used.

If inheritance is not set for an attribute, and the attribute's value is not set, then the default value is used.

Queues, Queue Sets and Queue Set References

IStream Publisher receives messages (or "requests") from **Queues**, interprets the XML request in the message, and then executes the instructions in the message.

IStream Publisher uses **Queue Sets** and **Queue Set References.**

A **Queue Set** is a logical grouping of queues based on the types of requests in the queues, in order to create a logical unit of work. Examples of Queue Sets include grouping the queues by

- line of business
- department or office location
- application or document type

Note: You need to configure at least one Queue Set in IStream Publisher. If you are using OpenJMS, a default Queue Set is automatically configured for you. If you are using MQ Series, you must configure the default Queue Set when adding a domain.

A **Queue Set Reference** (**QueueSetRef**) is assigned to each Worker. It contains the number of listeners and the name of the **Queue Set**. In this way, you are assigning groups of queues to specific workers.

By default, all workers will listen to the default queue set.

If you do not want the Worker to listen to the default Queue Set, you can assign a different Queue Set by selecting the **Set** check box for the `primaryQueueSet` and entering the name of the desire Queue Set.

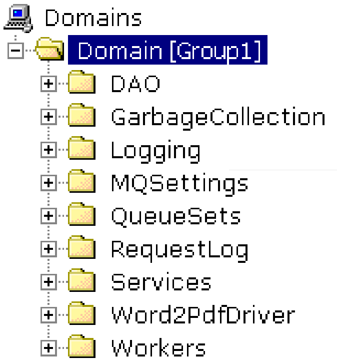
Changing Component Attribute Values

Administrators and Operators can use the **Configuration** module to view or change the attribute of a component.

Method: Change the value of a component's attribute

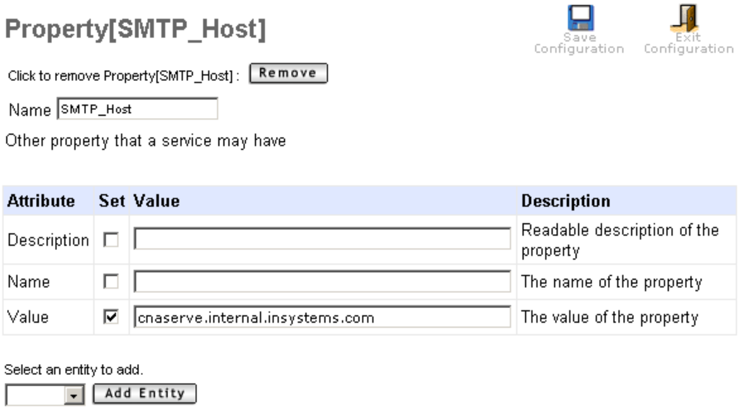
1. On the module menu, click **Configuration**.

   The **Configuration** tree is displayed on the left pane.



Domains
- Domain [Group1]
  - DAO
  - GarbageCollection
  - Logging
  - MQSettings
  - QueueSets
  - RequestLog
  - Services
  - Word2PdfDriver
  - Workers

2. Navigate through the tree to select the entity that contains the attribute you want to change.



Property[SMTP_Host]

Click to remove Property[SMTP_Host] :

Name SMTP_Host

Other property that a service may have

Attribute	Set	Value	Description
Description	□		Readable description of the property
Name	□		The name of the property
Value	☑	cnaserve.internal.insystems.com	The value of the property

Select an entity to add.

3. Select the **Set** check box beside an attribute to enter the attribute value in the **Value** field.

   Alternatively, leave the **Set** check box cleared. The value for the attribute is obtained from a parent entity that is higher in the entity structure, if the attribute's definition allows inheritance. If the value for all the parent entities is undefined, a default value is used.

   If you clear a **Set** check box, you cannot change the **Value** field, and the field changes to display the parent entity's value or its default value (which could be undefined).

You can only edit or define attribute values at the current level. If you need to change a parent value, you must open the parent entity.

4. To save your changes, click **Save Configuration** in the upper-right corner of the right pane.

Your changes are saved.

5. Execute the **Stop** Worker command, and then **Start**. This ensures your changes are applied.

Please note:

- Changes to Workers, QueueSets and QueueSetRefs are enabled only after issuing a `Start` command. All other changes require you to restart the `WorkerAgent` service in order for the changes to take effect.
- When saving a QueueSet in the IStream Publisher Console, ensure that you have entered all the required values correctly, because the system does not verify if any required values are missing or incorrect.

Managing Entity Instances

Administrators and Operators can perform the following functions in the **Configuration** module:

- *Creating a New Entity Instance* on page 59
- *Renaming an Entity Instance* on page 60
- *Deleting an Entity Instance* on page 61

Creating a New Entity Instance

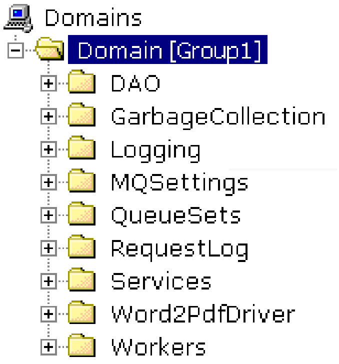
On each worker system, you can create copies of the same entity type: these are called *entity instances*. Each entity instance has the same attributes, but a unique name, and different attribute values. This allows you to customize the attribute values for each entity instance on each worker system. In this way, you are customizing how each component will perform on each worker system.

Note: You cannot create entities. You can only create entity instances.

Method: Create a new entity instance

1. On the module menu, click **Configuration**.

   The **Configuration** tree is displayed on the left pane.



Domains
Domain [Group1]
DAO
GarbageCollection
Logging
MQSettings
QueueSets
RequestLog
Services
Word2PdfDriver
Workers

2. In the configuration tree, navigate to the entity where you want to add an entity instance under. Note that you can only add entity instances to an entity that has an **Add Entity** button.
3. Select an entity from the drop-down list, then click **Add Entity**.

   The entity instances is created and its attributes are displayed on the page. The **Set** check box beside each attribute is clear and the value fields are read-only.
4. Select the **Set** check box beside each attribute you want to edit.

The **Value** fields become editable. (For more information about the **Set** check box, see step 3 in *Change the value of a component's attribute* on page 57.)

5. Enter the values into the **Value** fields.

6. To save your changes, click **Save Configuration** in the upper-right corner of the right pane.

The entity instances is saved and added to the configuration tree.

Note: You must change at least one attribute value or you will not be able to save the new entity.

Renaming an Entity Instance

When you add a new entity instance to the IStream Publisher console database, it is automatically named using a prefix string, followed by square brackets containing an instance identifier. When renaming an entity instance, you can change only the instance identifier portion of the name. You cannot change the prefix.

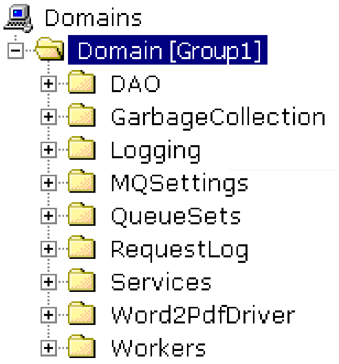
Note: You cannot rename entities. You can only rename entity instances.

Method: Rename an entity

Warning: Do not rename domains because this will cause IStream Publisher to stop working.

1. On the module menu, click **Configuration**.

The **Configuration** tree is displayed on the left pane.



1. In the configuration tree, navigate to the entity instance you want to rename.

The entity instances attributes and the **Name** field is enabled on the right side of the page.

2. In the **Name** field, enter the new name.

Note: You cannot use the following characters in the entity name: `' "  \ [ ]`

3. To save the new name, click **Save Configuration** in the upper-right corner of the right pane.

The new name is saved.

Deleting an Entity Instance

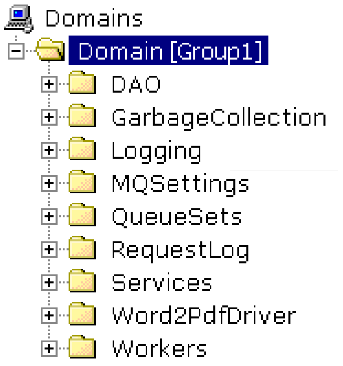
Deleting an entity instance removes it, its attributes, and its instances.

Warning: Delete only *entity instances*. Do not delete *entities*, because this can cause IStream Publisher to stop working.

Method: Delete an entity instance

1. On the module menu, click **Configuration**.

The **Configuration** tree is displayed on the left pane.



2. In the configuration tree, navigate to the entity instance that you want to delete.

The entity's instance's attributes are displayed on the right pane.

3. To delete the entity, click **Remove**.

A message asks you to confirm the deletion.

4. Click **OK** to confirm the deletion.

The entity instance is deleted.

Chapter 7

Managing Domains in the Console

This chapter describes:

- *The Manage Domain Module* on page 64
- *Adding a Domain* on page 66
- *Deleting a Domain* on page 68

The Manage Domain Module

A domain is a logical grouping of IStream Publisher components.

Administrators and Operators can use the **Manage Domain** module to issue commands to components while they are running, and to add new domains.

Warning: Do not rename domains because this will cause IStream Publisher to stop working.

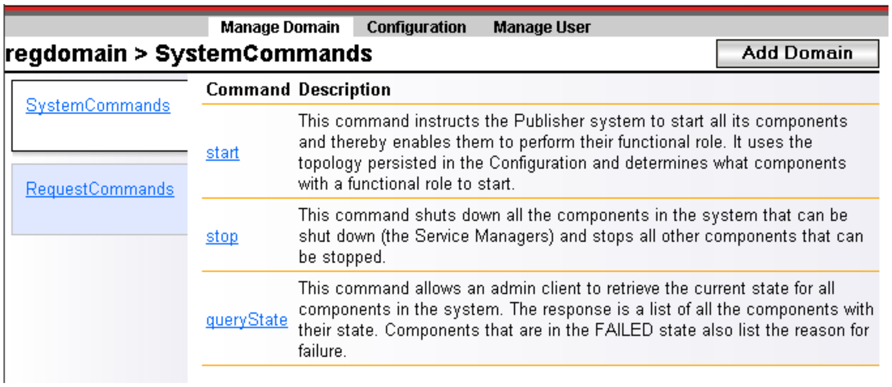
Viewing or Executing Commands

To view or execute a command for a component, you select a component and then select one of its commands.

Method: View or execute a command

1. On the module menu, click **Manage Domain**.
2. **If you have more than one domain configured:** click the **Domain Name** of the domain with which you want to work, then continue to the next step.

   If you have only one domain configured: continue to the next step.
3. The commands for the selected domain are displayed.



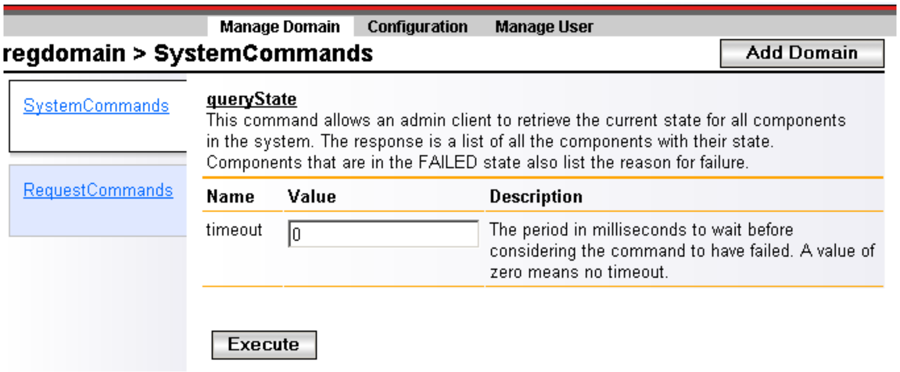
regdomain > SystemCommands

Command	Description
start	This command instructs the Publisher system to start all its components and thereby enables them to perform their functional role. It uses the topology persisted in the Configuration and determines what components with a functional role to start.
stop	This command shuts down all the components in the system that can be shut down (the Service Managers) and stops all other components that can be stopped.
queryState	This command allows an admin client to retrieve the current state for all components in the system. The response is a list of all the components with their state. Components that are in the FAILED state also list the reason for failure.

4. If you have more than one domain, and want to select another one to work with, select the domain from the drop-down list, then click **GO**.
5. On the left pane, select the type of command you want to execute, **System Commands** or **Request Commands**. By default, **System Commands** are selected.

   A list of corresponding **Command**s and their descriptions are displayed on the right pane. Each **Command** is a hyperlink.
6. Click the **Command** that you want to view or execute.

   A page is displayed for the specific **Command**, for example:



regdomain > SystemCommands

queryState

This command allows an admin client to retrieve the current state for all components in the system. The response is a list of all the components with their state. Components that are in the FAILED state also list the reason for failure.

Name	Value	Description
timeout	0	The period in milliseconds to wait before considering the command to have failed. A value of zero means no timeout.

7. From this page, you can view descriptions about the command and its values, or execute the command.

8. To execute the command, enter a valid **Value**, then click **Execute**. (Note that some commands have more than one **Value**.)

   The command is executed. (This may take a few seconds.) A message indicates whether the command was successful. A table lists the components, the result and state.

9. For details about the command you just executed, click the **Show/Hide XML result** link. This displays the result of the command in XML. Click the link again to hide the XML result.

10. To rerun the same command again, click **Rerun *command_name***, where ***command_name*** is the name of the command you have just run. (This is useful if you want to run the command using a different value.)

11. To return to the list of commands, click **Return to command list**, or click any component on the left side of the screen.

Note: To enable changes to Workers, QueueSets and QueueSetRefs, issue the `Start` command. All other changes require you to restart the `WorkerAgent` service in order for the changes to take effect.

Adding a Domain

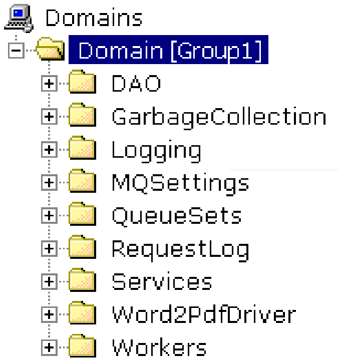
Complete the following procedure to add a domain to the IStream Publisher console.

You will need add a domain after performing a new installation of IStream Publisher.

Method: Add a domain

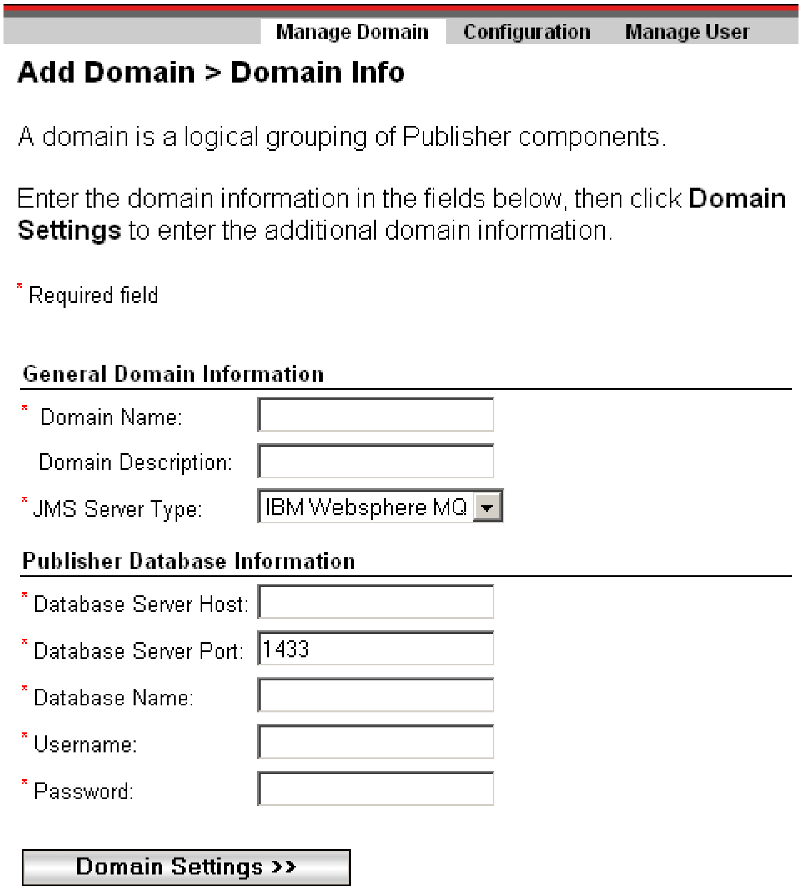
1. On the module menu, click **Configuration**.

The **Configuration** tree is displayed on the left pane.



2. Click **Add Domain**.

The **Add Domain** page opens.



Add Domain > Domain Info

A domain is a logical grouping of Publisher components.

Enter the domain information in the fields below, then click **Domain Settings** to enter the additional domain information.

* Required field

General Domain Information

* Domain Name:	
Domain Description:	
* JMS Server Type:	IBM Websphere MQ

Publisher Database Information

* Database Server Host:	
* Database Server Port:	1433
* Database Name:	
* Username:	
* Password:	

3. Complete the required fields in this page and the **Domain Settings** page. (An asterix indicates a required field.) For field descriptions, see *Step B: Obtain Domain Information* on page 110.

4. After entering all the values, click **Add Domain**.

   The domain is added, and a page listing the current domains opens.

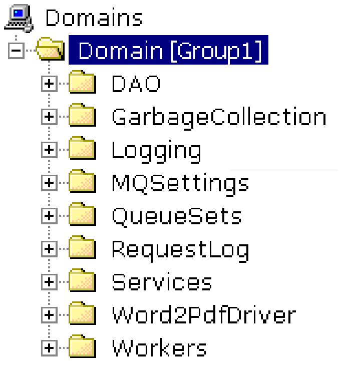
Deleting a Domain

Deleting a domain removes its configuration information from the IStream Publisher console database.

Method: Delete a domain

1. On the module menu, click **Configuration**.

   The **Configuration** tree is displayed on the left pane.

2. In the configuration tree, select the domain that you want to delete.

   The domain’s attributes are displayed on the right pane.

3. Click **Remove**.

   A message asks you to confirm the deletion.

4. Click **OK** to delete the domain.

   The domain is deleted.

Chapter 8

Advanced Configuration

This chapter describes the following advanced functions:

- *Adding Workers* on page 70
- *Removing Workers* on page 71
- *Other Configuration* on page 72

Adding Workers

This section describes how to add Workers to the system. Typically, you will want to add Workers when you need to increase the throughput of IStream Publisher.

Every IStream Publisher component's performance is effected by various factors, such as the operation performed by the component, the hardware and software installed on the system, and so on.

Method: Add a Worker

1. Install new Workers on separate computers. Use the same parameters used for the Console database and the same IStream Publisher Domain Name used for the IStream Publisher installation.
2. Launch all IStream Publisher components by starting Windows Services or executing start-up batch files.
3. In the IStream Publisher Console, click **Manage Domain**.

   When you run the `QueryState` command, the new Worker should appear in the result.

Note: If a *Selected system is not available* error message appears, you need to verify the Worker's configuration and availability (make sure it has been started and is running).

Removing Workers

1. Stop the selected components (by stopping Windows Services or closing the process windows).
2. In the IStream Publisher console, click **Configuration**.
3. Open the `Domains` *`domain_name`*`/Workers` folder.
4. Open the Worker you want to remove, then click **Remove**.

Other Configuration

This section describes other, assorted configuration information and procedures.

Configure JNDI (LDAP)

These settings are optional.

Configuration parameters	Configuration Path in the Console Database
JNDI Factory Class	`.../JNDISettings/Property[java.naming.factory.initial]/Value` **Example:** `com.sun.jndi.ldap.LdapCtxFactory`
JNDI URL	`.../JNDISettings/Property[java.naming.provider.url]/Value` **Example:** `ldap://edelivery:389/o=internal.edelivery`
JNDI Principal	`.../JNDISettings/Property[java.naming.security.principal]/Value` **Example:** `uid=edelivery,ou=Special Users,o=internal.edelivery`
Credentials	`.../JNDISettings/Property[java.naming.security.credentials]/Value` **Example:** `password`
Referral	`.../JNDISettings/Property[java.naming.referral]/Value` **Example:** `throw`

Encrypted Credentials

See *SDK - Encrypted Credentials* on page 185 of the *IStream Publisher Interface Reference Guide*.

Default Password Guidelines

To change the default password guidelines, edit the `XRMConfig.XML` file on the Console machine.

Log Levels

You can configure the `logLevel` parameter for the agents to control the amount of logging that takes place. Because the amount of logging occurring affects performance, this setting should be carefully monitored.

The `logLevel` types are:

- DEBUG
- INFO

- WARN
- ERROR
- FATAL

These log levels are listed in order of severity. The higher the log level, the more log messages will be logged. The default level is `INFO`.

Important: Ensure that the letters in the `LogLevel` are in upper-case.

Changing the LogLevel

To change the logging level, complete the following steps:

1. In the IStream Publisher console, click **Configuration.**
2. Open the `Domains[`*`domain_name`*`]/Logging` folder.
3. Enter a **Value** for the **logLevel** attribute.
4. Click **Save Configuration**.

Assembly Engine Timeout

Complete the following steps to assign the Assembly Engine timeout value:

1. In the IStream Publisher console, click **Configuration**.
2. Open the `Domains/Domain[`*`domain_na`*`me]/Services/Service [`*`GenerationService`*`]/Property[NativeSettings]` folder.
3. In the **Value** attribute, include the `-timeout TTT` option, where `TTT` is the timeout in milliseconds. Ensure that this value is less than the `timeout` value in `Service [`*`GenerationService`*`]`.

Changing the OpenJMS Port Value

Complete each of the following procedures in this order to change the OpenJMS port number. The default value is `1099`.

Step A: Update the OpenJMS Configuration

1. Open the following file:

   `IStream Publisher install folder``\openjms\config\`
   `openjms.xml`

2. Locate the following line:

   ```
   <ServerConfiguration host="host_name"
   embeddedJNDI="true" />
   ```

3. Immediately below this line, add the following line:

   ```
   <RmiConfiguration embeddedRegistry="true"
   registryPort="port_value" registryHost="host_name"/>
   ```

Step B: Update the Console Configuration

1. In the Console, click the **Configuration** tab.
2. Navigate to the following property

   ```
   domain_name/JNDISettings/
   Property[java.naming.provider.url]
   ```

3. In the **Value** attribute, change `1099` to the port value you specified in *Step A: Update the OpenJMS Configuration* on page 73.
4. Click **Save Configuration**.
5. Restart all IStream Publisher services on all workers.

Step C: Update the Test Console Configuration

1. Open the Test Console. (For details, see *Starting the Test Console* on page 76.)
2. Press **Ctrl-Shift-S**, then click **EDIT**.
3. Locate the following line in the text section of the **TCSetup** dialog box:

   ```
   java.naming.provider.url=rmi://host_name:1099/
   ```

4. Change `1099` to the value you specified in *Step A: Update the OpenJMS Configuration* on page 73.
5. Click **Save** and then **Test**.

   The Test Console should report a successful connection to the OpenJMS server.

Chapter 9

The Test Console

This chapter describes the Test Console, an application you can use to create, edit and submit XML requests to IStream Publisher, and in doing so, test that your requests function correctly.

This chapter describes:

- *Installing and Starting the Test Console* on page 76
- *Setting up the Test Console* on page 77
- *Opening Requests* on page 79
- *Working with Requests* on page 82
- *Using the Load Runner* on page 85
- *Working with Queues* on page 92
- *Performing Administrative Functions* on page 96
- *Viewing the DMS* on page 97
- *Issuing SQL Commands* on page 98

Installing and Starting the Test Console

The Test Console is an application that allows you to create, save and manually submit requests to IStream Publisher, as well as monitor the queues. In doing so, you are verifying that your requests and IStream Publisher are working properly.

Installing the Test Console

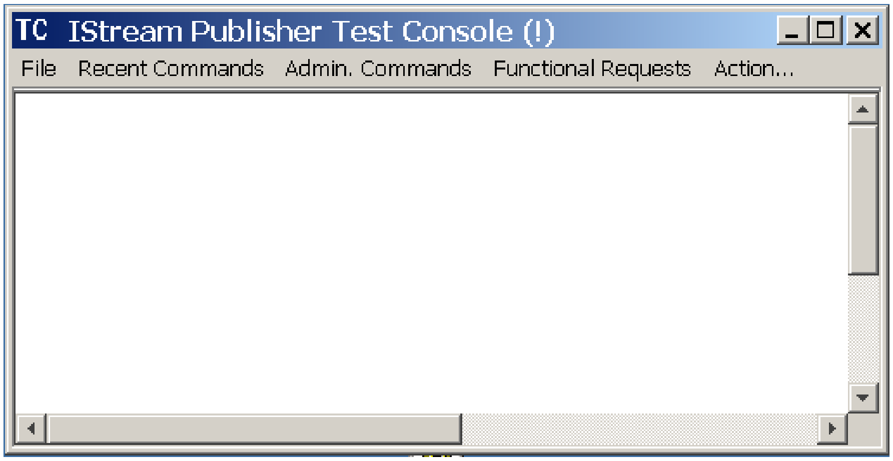
You can select to install the Test Console during the IStream Publisher installation process. In the **IStream Publisher Core Components** screen, ensure that **Install Test Console** is selected.

Starting the Test Console

To start the Test Console, double-click the `startTestConsole.bat` file located in:

`\`*`IStream Publisher_install_folder]`*`\bin\`

The Test Console opens, with the main edit screen displayed.



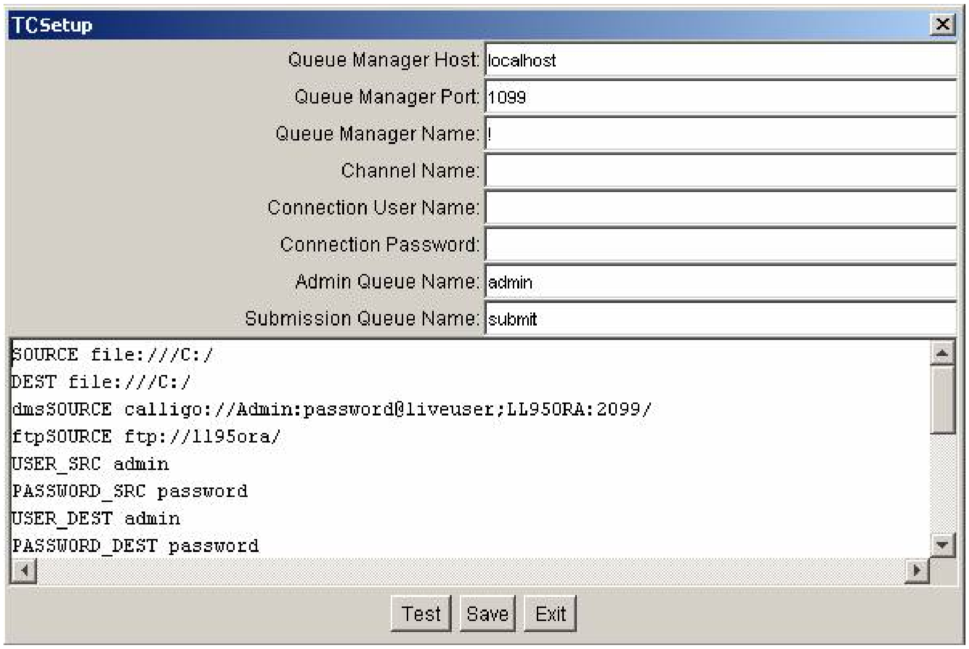
Using the Test Console Menus

After you start and configure the Test Console, you will see the following menus:

- **File** – this is the main menu; you use it to load, save, monitor and submit requests, browse and monitor queues and perform other functions
- **Recent Commands** – this menu contains a list of commands you have issued during your Test Console session
- **Admin Commands** – this menu allows you to perform various administrative tasks: for details, see *Performing Administrative Functions* on page 96
- **Functional Requests** – this menu contains all your executed requests
- **Action** – this menu contains the commands for executing and stopping requests

Setting up the Test Console

Use the **Setup** module to set up and test the Test Console.



Queue Manager Host: localhost
Queue Manager Port: 1099
Queue Manager Name: !
Channel Name:
Connection User Name:
Connection Password:
Admin Queue Name: admin
Submission Queue Name: submit

```
SOURCE file:///C:/
DEST file:///C:/
dmsSOURCE calligo://Admin:password@liveuser;LL95ORA:2099/
ftpSOURCE ftp://ll95ora/
USER_SRC admin
PASSWORD_SRC password
USER_DEST admin
PASSWORD_DEST password
```

Method: Set up the Test Console

1. Click **File > Setup**, or press **Ctrl-Shift-S**.
2. In the **Setup** dialog, enter the following values:
   - **Queue Manager Host** – for MQ series and OpenJMS
   - **Queue Manager Port** – for MQ series and OpenJMS
   - **Queue Manager Name** – for MQ series, enter the name; for OpenJMS, enter an exclamation mark (!)
   - **Channel Name** – for MQ series only
   - **Connection User Name** – for MQ series only
   - **Connection User Password** – for MQ series only
   - **Admin Queue Name** – for MQ series and OpenJMS; the default value is `admin`
   - **Submission Queue Name** – for MQ series and OpenJMS; the name of the submission queue in any of your queue sets, this is typically `submit`
3. To test the connection to the JMS queues, click **Test.** If the test is successful then the setup settings are saved. If the test fails, the system reverts to the previous settings when you close the **Setup** dialog.
4. To enter variables, complete the following steps:

a. Click **Edit**. The **Edit** button changes to a **Save** button and an edit pane opens.

b. To create a new variable, use the following format on a new line:

```
<variable name> <variable value>
```

c. To use the variable, place the text `${<variable name>}` anywhere in the request and the Test Console will replace it with its value before submitting the request.

d. To remove a variable, remove it from this edit pane, save the changes and then restart the Test Console.

5. To save your settings, click **Save**.

6. When you are done, click **Exit** to close the **Setup** dialog,

Opening Requests

There are different ways to open requests. This section describes:

- *Loading Requests* on page 79
- *Finding or Replacing Text* on page 82
- *Opening Functional Requests* on page 80
- *Creating New Requests* on page 81

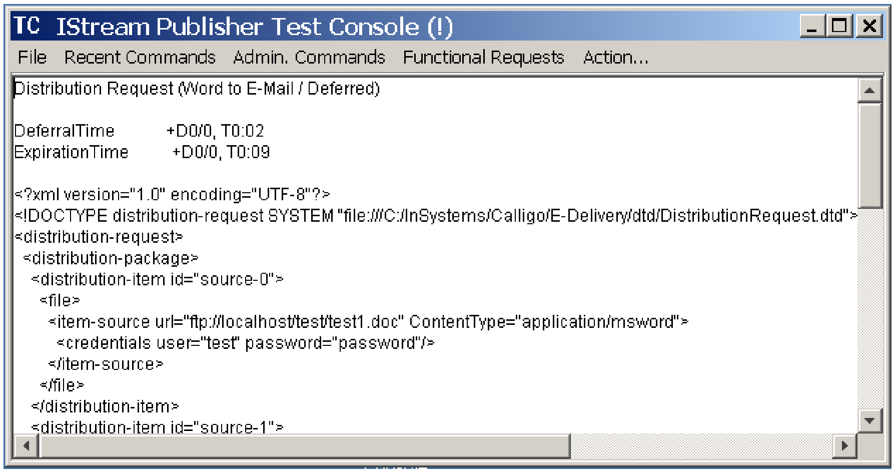
Loading Requests

If a submission request file already exists, you can load it into the Test Console

Method: Load an existing request

1. Click **File** > **Load**, or press **Ctrl-L**.
2. Navigate to the folder containing the request file you want to load.
3. Once you have located the file, double-click it.

   The file is now loaded into the Test Console.



```
Distribution Request (Word to E-Mail / Deferred)

DeferralTime        +D0/0, T0:02
ExpirationTime       +D0/0, T0:09

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE distribution-request SYSTEM "file:///C:/InSystems/Calligo/E-Delivery/dtd/DistributionRequest.dtd">
<distribution-request>
 <distribution-package>
  <distribution-item id="source-0">
   <file>
    <item-source url="ftp://localhost/test/test1.doc" ContentType="application/msword">
     <credentials user="test" password="password"/>
    </item-source>
   </file>
  </distribution-item>
  <distribution-item id="source-1">
```

Tip: You can load multiple files by using Ctrl-click to select multiple entries in the **Open File** dialog. The Test Console will save all opened files to the **Functional Requests** menu and load the contents of the last file from the selection into the edit pane.

Opening Recently Run Commands

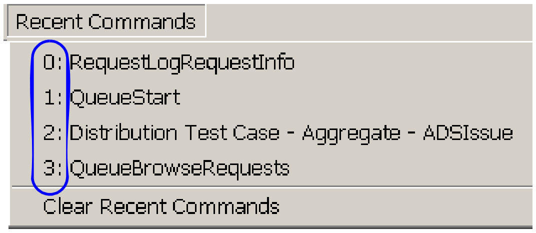
You can open a recently run command to view, edit or run it.

Method: Open a recently run command

1. Click **Recent Commands**.
2. From the list, select the command that you want to work with.

Tip: You can use the **Alt-R-<*character*>** keystroke combination to quickly load a recent command, where <*character*> is 0 to 9 or A to E and corresponds to the character assigned to the command in the menu.

In the following example, you can type **1** to select the **QueueStart** command:



The contents of the command are displayed.

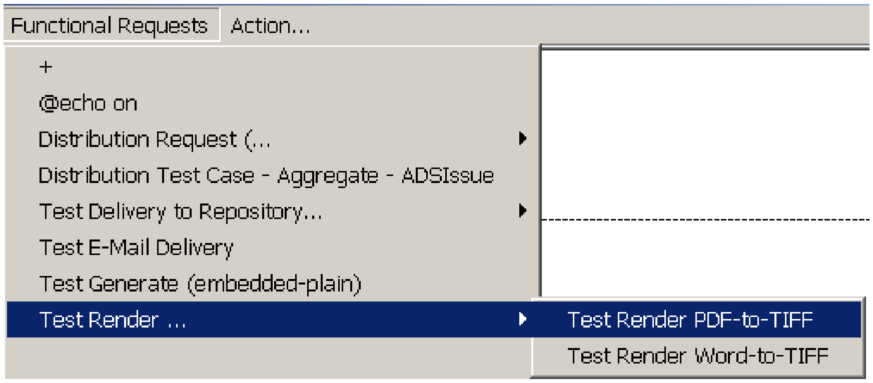
Tip: To clear the list of recent commands, click **Recent Commands > Clear Recent Commands**.

Opening Functional Requests

You can open a previously executed request to view, edit or run it.

Method: Open a functional request

1. Click **Functional Requests**.
2. Select a request from the menu. Note that the requests are grouped in the menus based on the text in the first line of the request, for example:


Functional Requests
Action...
+
@echo on
Distribution Request (...
Distribution Test Case - Aggregate - ADSIssue
Test Delivery to Repository...
Test E-Mail Delivery
Test Generate (embedded-plain)
Test Render ...
Test Render PDF-to-TIFF
Test Render Word-to-TIFF

The contents of the request are displayed.

Creating New Requests

You can create a new request in the Test Console, and then save it for future use.

Method: Create a new request

1. Ensure the edit pane is blank.
2. Enter the new request using the following format:

   `<title>`

   `<JMS properties>`

   `<request>`

   where

   - `<title>` is used by the Test Console to uniquely identify the request
   - `<JMS properties>` are a list of JMS properties (one on each line) that can be set on the message (for example, `JMSPriority 6`) using the format `<property name> <property value>`
   - `<request>` is the request XML
3. Save the new request to the Test Console by executing it: for details on saving to a file system, see *Saving Request Files* on page 83.

Working with Requests

This section describes:

- *Running a Request* on page 82
- *Stopping a Request* on page 82
- *Finding or Replacing Text* on page 82
- *Saving Request Files* on page 83
- *Printing Request Files* on page 83
- *Removing Commands and Requests from Menus* on page 84

Running a Request

After you have loaded a request, you can run it.

Method: Run a request

1. Ensure you have loaded a request.
2. Click **Action** > **Execute**, or press the **F9** key.

   The request is run. The **Results** window opens and displays the message that was sent, and any reply that is returned.

Stopping a Request

If a request is running, you can stop it if necessary. However, note that stopping a request may not always stop IStream Publisher from processing it.

Note: If IStream Publisher has already begun processing the request, then stopping it in the Test Console only stops the Test Console from waiting for a reply. If IStream Publisher has not yet picked up the request from the submission queue, then stopping the request will also delete the request from the queue.

Method: Stop a running request

- Click **Action** > **Stop**, or press the **Esc** key.

  The request stops.

Finding or Replacing Text

You can find or replace text within a request.

Method: Find or replace text in a request

1. After loading a request, click **File > Find,** or click **Ctrl-F**.
2. If you only want to search or replace text within a certain area, select the relevant section of text to search.
3. In the **Find/Replace** dialog, enter the **Text to find**.
4. Optionally, in the **Replace with** field, enter the text to replace the text you are searching for.
5. Select whether the **Text to find** is **Case sensitive**.
6. If you want to find or replace from the end of the document to the beginning, select **Reverse search.**
7. If you had selected only a certain area of the text to find or replace, (in step 2), select **Search only selected text** to find or replace text only within the selected area.

Saving Request Files

If you want to back up a request file, or create a copy of it on your file system, you can save it.

Method: Save a request file

1. Click **File > Save to File**, or press **Ctrl-S.**
2. By default, the **File name** is made up of the letters in the first line of the file. Change the **File name** if necessary, then click **Save.**

   The file is saved.

Method: Export all requests

You can export all **Administrative Commands** and **Functional Requests** to a folder.

1. Click **File > Export,** or press **Ctrl-Shift-E**.
2. Select (or create) a folder to export the files to**.**
3. Click **Save**.

   The commands and requests are exported to the selected folder.

Printing Request Files

You can print a loaded request file.

Method: Print a request file

1. Click **File > Print**, or press **Ctrl-P.**
2. In the **Print** dialog, enter a **Print range** (if applicable) and the **Number of Copies** to print.
3. Click **OK** to print.

Removing Commands and Requests from Menus

You can remove a command or request from the **Recent Commands**, **Admin Commands**, and **Functional Requests** menus.

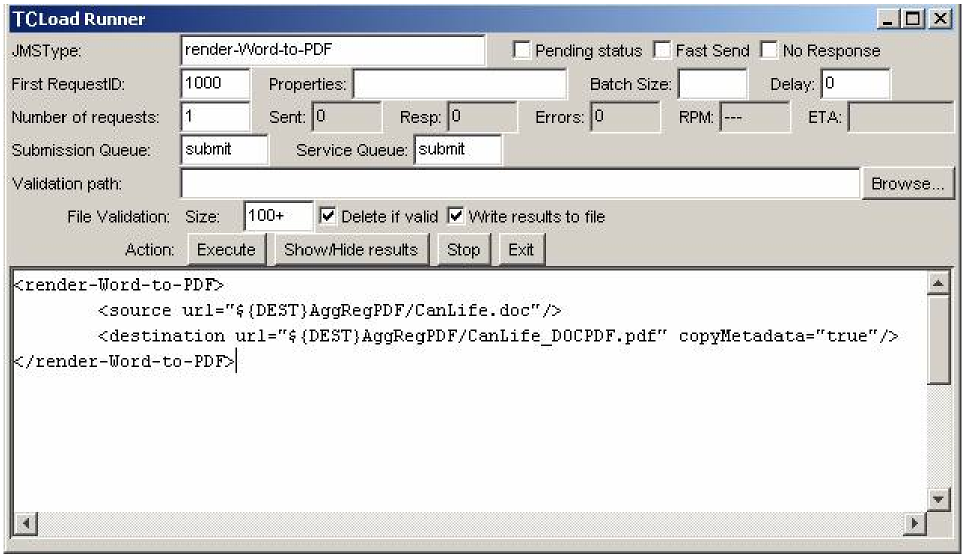
Method: Remove a command or request

1. Open the request or command you want to remove.
2. Click **File** > **Remove Request**, or press **Ctrl-Shift-R**.

   The item is removed from one or more of the **Recent Commands**, **Admin Commands**, and **Functional Requests** menus.

Using the Load Runner

You can use the Load Runner to send batches of requests.



JMSType: render-Word-to-PDF ☐ Pending status ☐ Fast Send ☐ No Response

First RequestID: 1000 Properties: Batch Size: Delay: 0

Number of requests: 1 Sent: 0 Resp: 0 Errors: 0 RPM: --- ETA:

Submission Queue: submit Service Queue: submit

Validation path:

File Validation: Size: 100+ ☑ Delete if valid ☑ Write results to file

```
<render-Word-to-PDF>
        <source url="${DEST}AggRegPDF/CanLife.doc"/>
        <destination url="${DEST}AggRegPDF/CanLife_DOCPDF.pdf" copyMetadata="true"/>
</render-Word-to-PDF>
```

Method: Use the Load Runner

1. Click **File > Load Runner**, or press **Ctrl-R**.
2. In the **Load Runner** dialog, make the following entries and selections:
   - Enter the **JMSType** of the message.
   - Select the **Pending status** checkbox to send the request directly to the Service queue.The request will not be logged in the database. If selected, change the **Submission Queue** to point to a service queue of a queue set.
   - Select the **Fast Send** checkbox to send the request in batches containing a maximum of 200 requests. This is useful for testing the maximum messaging throughput of some components.
   - Select the **No Response** checkbox to force the Test Console to keep the `JMSReplyTo` property empty in the requests and not wait for the responses. This may be useful if you want to send a large batch, close the application and later collect the execution results from the IStream Publisher database.
   - Enter JMS **Properties** of the message using the following format:

     `property_name=property_value/property_name=property_value...`

     Example:

     `JMSPriority=4`

     `JMSPriority=2/Requestor=Me`

- Enter the **Submission Queue** and **Service Queue** for the request.

3. If you want to set the `RequestIDs` of the requests sent:

- Enter a number or string in the **First RequestID** field:
  - If you enter a *number*, the messages IDs will be numbered incrementally up from that number. For example, if you enter **`30`**, the request ID for the first request will be `30`, followed by `31`, `32` and so on.
  - If you enter a *string*, the message IDs will be numbered incrementally from 0 using that string as a prefix. For example, if you enter **`abc`**, the request ID for the first request will be `abc0`, followed by `abc1`, `abc2` and so on.

4. If you want to send requests in batches:

a. Enter the number of requests to be sent in each batch in the **Batch Size** field.

b. Enter the total **Number of requests** to be submitted.

c. Enter the number of seconds to wait between each batch submission in the **Delay** field.

5. If you want to validate the output file created by the request:

a. Click **Browse** to select a **Validation path**.

This is required only if you want to verify that a required file is in place after a request has completed successfully and that the file has a valid size. The path should match that of one of the destinations from the request and can be a Windows path (UNC, mapped drive, and so on) or a DMS URL (for example, `calligo://user:password@connection;server/path/filename$M{id}`).

Note: We recommend validating on a file system because any additional operations with the DMS can affect performance.

b. In the **File Validation** section, enter the expected **Size** (in bytes) of the output file. You can enter:

- a range (`100-300`)
- a minimum value (`100+` indicates the file size must be at least 100 bytes
- an exact value, for example, `100` indicates the file size must be exactly 100 bytes to be valid

Note: We recommend entering a range or minimum value.

c. Select the **Delete if valid** checkbox to have IStream Publisher automatically delete the file if it is valid.

d. Select the **Write results to file** checkbox to log the submission results to *`IStream Publisher_install_folder`*`\bin\runner.log`.

6. In the **Action** section, click **Execute** to issue the command.

The results of the command are displayed in the following read-only fields:

- **Sent** – the number of messages sent
- **Resp** – the number of responses received from IStream Publisher
- **Errors** – the number of responses that were errors
- **RPM** – the number of responses received per minute; this value is accurate only if the number of requests sent is large
- **ETA** – the estimated completed time for all the responses to be returned

7. To hide the results, click **Show/Hide** results. Click this button again to re-display the results.

8. To stop the command before it has finished running, click **Stop**. Load Runner deletes any messages from the specified service queue that IStream Publisher has not picked up for processing.

9. When you are done, click **Exit** to close the **Load Runner**.

Please note:

- Clicking **Exit** while a batch is running will **Stop** the command.
- Load Runner will try to pick up data from the main edit pane of the Test Console when it is launched. If you open Load Runner and see an old request, press **Exit** and reopen Load Runner to refresh it.
- You can use the **Number of Requests** field to set an optional queue size limit. For example, if you set this field to `1000/100`, Load Runner will send 100 requests and will only send one more request for each response received after that (up to 1,000). This limits the number of messages that will wait in the queue at any given time during execution. In this example, no more than 100 messages will be in the queue.
- If after a batch has started and not all the requests have been submitted to the queue (for example, because the **Batch Size** and **Delay** parameters have been set), you can change the: request contents, **Number of requests**, **Batch size**, **Delay** and **JMS properties**, then click **Update**. The remaining requests in the batch will be sent with the new parameters.

Using Macros in Requests

With the Load Runner, you can use the following macros anywhere in a request. Typically, they are used for the destination URLs to generate unique URLs for each request in the batch.

Example Macro

```
<destination url="file://server/share/Test$M{fid}/$M{/1000}/
$M{0-999/100}/test$M{id}.doc"/>
```

A batch executed using this example saves the results on a shared file system, using the following hierarchy:

Batch folder > **Subfolder** > **Sub-subfolder** > **Request**

The details of this hierarchy are as follows:

1. **Batch folder** – Corresponds to `Test$M{fid}` in the example macro.

   Each batch has its own folder, which in turn, contains subfolders and sub-subfolders.

2. **Subfolder** – Corresponds to `$M{/1000}` in the example macro.

   This subfolder contains a total of 1,000 documents. If there are more than 1,000 requests in the batch, then a new subfolder is created.

   Each subfolder contains 10 sub-subfolders.

3. **Sub-subfolder** – Corresponds to `$M{0-999/100}` in the example macro.

   Each sub-subfolder contains 100 requests.

4. **Request** – Corresponds to `test$M{id}.doc` in the example macro.

   Each request has a unique filename based on the `RequestID`.

Macro Syntax

The **Macro Syntax** column describes the syntax and values used in the macro.

The **Description** column describes the values that replace the macro when it is run.

Macro Syntax	Description
`$M{id}`	Returns the `RequestID` of the current request.
`$M{fid}`	Returns the value of the text in the `First RequestID`.
`$M{RND}`	Returns a random number from 1 to 999999 inclusive.
`$M{RNDX-Y}` where `X` and `Y` are numbers	Returns a random number in the range of `X` to `Y` (inclusive).

`$M{RNDX-Y|00…}` where `X` and `Y` are numbers, and `00…` represents the number of digits in the values produced	Returns a random number from `X` to `Y`.
	The number of digits in the values produced will equal the number of `0`’s entered in the macro. For example, if you have two zeros (`00`) in the macro, then the values produced will have two digits only.
	Example:
	`$M{RND5-800|000}`
	could produce these random values:
	`047`
	`357`
	`018`

Macro Syntax	Description

`$M{X-Y}` where `X` and `Y` are numbers	Returns a sequence of numbers from `X` to `Y`.
	The sequence repeats after `Y` is reached.
	Example:
	`$M{1-3}`
	would produce these values:
	`1`
	`2`
	`3`
	`1`
	`2`
	`3`

`$M{X,Y,Z,…}` where `X`,`Y`,`Z` are numbers	Returns a specific series of numbers `X`,`Y`,`Z` and so on.
	The first value is `X`, the second value is `Y`,
	the third value of `Z`, and so on.
	The sequence repeats after the `Z` is reached.
	Example:
	`$M{10,20,30}`
	would produce these values:
	`10`
	`20`
	`30`
	`10`
	`20`
	`30`

`$M{/X}` where `X` is a number	Returns a series of numbers that repeat `X` times.
	The first `X` values are 0,
	the next `X` values are 1,
	the next `X` values are 2,
	and so on.
	Example:
	`$M{/2}`
	would produce these values:
	`0`
	`0`
	`1`
	`1`
	`2`
	`2`

Macro Syntax	Description
`$M{X-Y/Z}` where `X`, `Y` and `Z` are numbers	Returns the series (`X` through `Y`) divided by `Z`. The resulting number is then rounded down. The sequence repeats after `Y` is reached. **Example:** `$M{1-6/3}` would produce these values (comments are in brackets): `0` (1/3 rounded down) `0` (2/3 rounded down) `1` (3/3) `1` (4/3 rounded down) `1` (5/3 rounded down) `2` (6/3)

Macro Syntax	Description
`$M{X-Y/Z+W}` where `W`, `X`, `Y` and `Z` are numbers	Returns the series (`X` through `Y`) divided by `Z`, then adds `W` to the result. The sequence repeats after `Y` is reached. **Example:** `$M{1-6/3+1}` would produce these values (comments are in brackets): `1` (1/3 rounded down + 1) `1` (2/3 rounded down + 1) `2` (3/3 + 1) `2` (4/3 rounded down + 1) `2` (5/3 rounded down + 1) `3` (6/3 + 1)
`$M{X-Y/Z+W|00…}` where `W`, `X`,`Y` and `Z` are numbers, and `00…` represents the number of digits in the values produced	Returns the series (`X` through `Y`) divided by `Z`, with `W` added to the result The number of digits in the values produced will equal the number of `0`'s entered in the macro. For example, if you have two zeros (`00`) in the macro, then the values produced will have two digits only. The sequence repeats after `Y` is reached. **Example:** `$M{1-6/3+1|00}` would produce these values (comments are in brackets): `01` (2 digits, 1/3 rounded down + 1) `01` (2 digits, 2/3 rounded down + 1) `02` (2 digits, 3/3 + 1) `02` (2 digits, 4/3 rounded down + 1) `02` (2 digits, 5/3 rounded down + 1) `03` (2 digits, 6/3 + 1)

Working with Queues

This section describes:

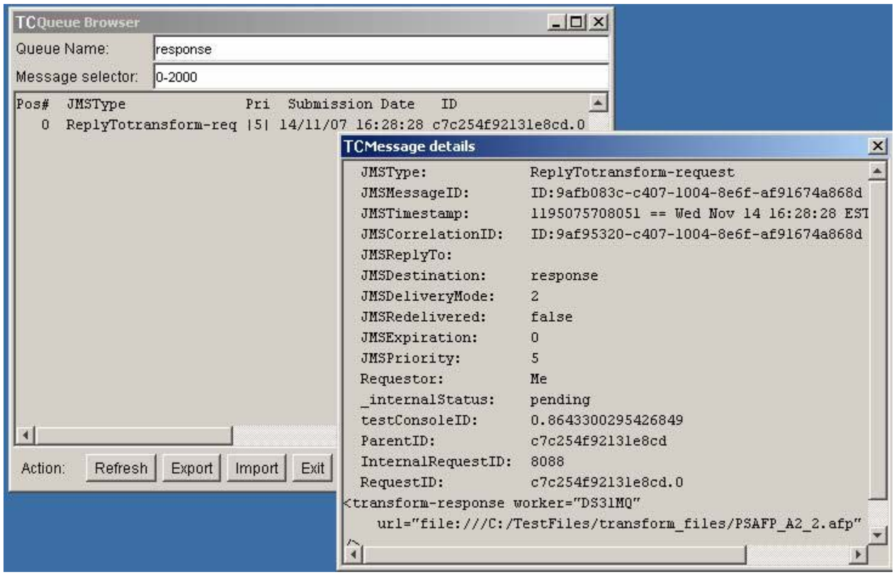
- *Browsing Queues* on page 92
- *Monitoring and Creating Queues* on page 94

Browsing Queues

You can use the Queue Browser to browse messages within a queue.

Method: Browse a queue

1. Click **File > Queue Browser**, or press **Ctrl-B.**



Queue Name: response

Message selector: 0-2000

```
Pos#  JMSType               Pri  Submission Date   ID
   0  ReplyTotransform-req |5| 14/11/07 16:28:28 c7c254f92131e8cd.0
```

```
 JMSType:               ReplyTotransform-request
 JMSMessageID:          ID:9afb083c-c407-1004-8e6f-af91674a868d
 JMSTimestamp:          1195075708051 == Wed Nov 14 16:28:28 EST
 JMSCorrelationID:      ID:9af95320-c407-1004-8e6f-af91674a868d
 JMSReplyTo:
 JMSDestination:        response
 JMSDeliveryMode:       2
 JMSRedelivered:        false
 JMSExpiration:         0
 JMSPriority:           5
 Requestor:             Me
 _internalStatus:       pending
 testConsoleID:         0.8643300295426849
 ParentID:              c7c254f92131e8cd
 InternalRequestID:     8088
 RequestID:             c7c254f92131e8cd.0
<transform-response worker="DS31MQ"
    url="file:///C:/TestFiles/transform_files/PSAFP_A2_2.afp"
/>
```

2. In the **Queue Browser** dialog, complete any of the following steps:
   - enter the name of the queue you want to brows in the **Queue Name** field
   - in the **Message selector** field:
     - enter a single number to display a specific message: for example, entering `4` will display the fourth message in the queue
     - enter a range of messages to display, for example `0-100`
     - enter a JMS message selector, for example, `RequestID LIKE 'abc%'`
   - double-click any message to view its details
   - to refresh the view, click **Refresh**
3. You can import a message provided it is in the correct format.

   a. Click **Import.**

   - **b.** Navigate to the file to import, then double-click it to import it. If successful, a list of all the imported messages appears.

4. To export a message:

   - **a.** Highlight the **Pos #** of the message you want to export.
   - **b.** Click **Export**. If successful, a message indicates that the message(s) have been saved.

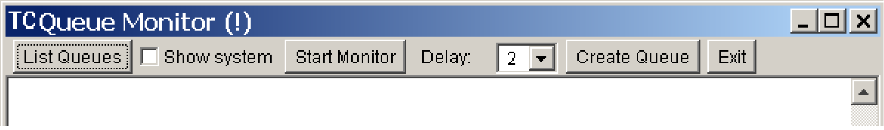
5. When you are done, click **Exit** to close the **Queue Browser.**

Monitoring and Creating Queues

Use the Queue Monitor to list, monitor and create queues.

Method: Monitor a queue

1. Click **File** > **Queue Monitor**, or press **Ctrl-Q.**



2. In the **Queue Monitor** dialog, complete any of the following steps:
   - to list the queues, click **List Queues**
   - to also list the system queues, select the **Show system** checkbox (for MQ Series only), then click **List Queues** to refresh the list
   - to set the interval in seconds that the queues are monitored, select a value from the **Delay** drop-down list
3. To enter a specific queue to monitor:

   a. Ensure that the **Start Monitor** button is displayed. (This indicates you have stopped the monitor.)

   b. Type the queue name directly into the **Queue Monitor** dialog.
4. To begin monitoring the listed queues, click **Start Monitor.**

   The queue messages are displayed.
5. To browse the messages, double-click the queue name, or right-click the message list and select **Browse messages**. The **Queue Browser** opens.
6. For MQ Series only, you can create a quick test message to submit to the queue:

   a. Right-click in the message list and select **Submit test message**.

   b. Enter a message then click **Submit.**
7. To create a queue, click **Create Queue**.

Important: We recommend using the default values in all fields:

- **Queue Name** – the name for this queue

The remaining fields are for MQ series only:

- **Description** – an optional description for this queue
- **Maximum Depth** – the maximum number of messages for this queue
- the **Maximum Message Length** in bytes
- the **Default Priority**
- the **Default Persistence** – values are **Non-Persistent** or **Persistent**

- the **Default Sequence** – values are **Priority** or **FIFO**
- the **Usage** type – values are **Normal** or **Transmission**

8. When you are done, click **Exit** to close the **Queue Monitor.**

Performing Administrative Functions

You can use the **Admin Commands** menu to perform the following administrative tasks:

- **SystemStop** – stops all IStream Publisher workers; equivalent to the `stop` IStream Publisher Console system command
- **SystemStart** – starts all IStream Publisher workers; equivalent to `start` IStream Publisher Console system command
- **SystemQueryState** – queries the state of all IStream Publisher workers; equivalent to `queryState` IStream Publisher Console system command
- **SystemCancelReques**t – cancels a request with the given options; equivalent to `cancelRequest` IStream Publisher Console request command

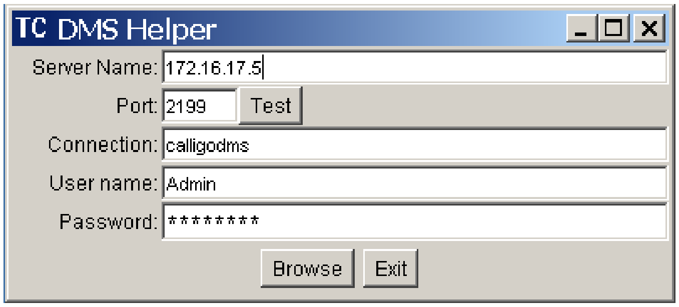
Important: By default, submitting these commands from Test Console will submit the messages to the admin queue that is defined in the Test Console setup (under the **Admin Queue Name**). These commands will therefore be processed by any domain listening on the this admin queue.

You can view the syntax for these XML requests in the following file:

`IStream Publisher_install_folder``/dtd/Administration.dtd`

Viewing the DMS

Use the DMS Helper to view and test the DMS.



TC DMS Helper

Server Name: 172.16.17.5

Port: 2199

Connection: calligodms

User name: Admin

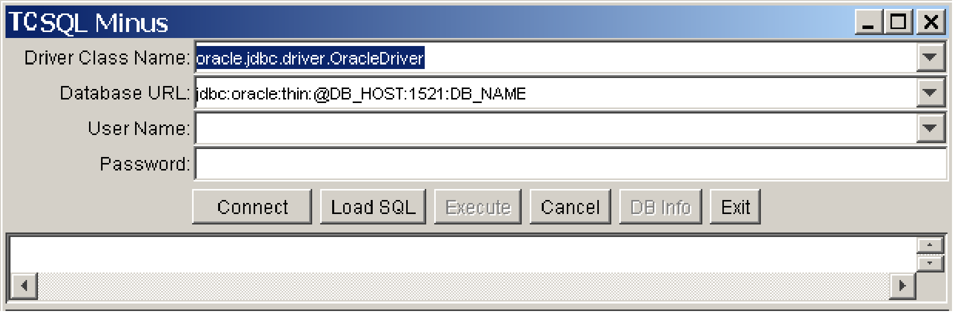
Password: ********

Method: View and test the DMS

1. Click **File > DMS Helper**, or press **Ctrl-D**.
2. In the **DMS Helper** dialog, enter or change the following values:
   - **Server name –** the host name
   - the **Port** number used by the DMS
   - the name of the **Connection**
   - the **User name** and **Password**
3. Click **Test** to test the connection to the DMS.
4. Click **Browse** to browse the contents of the DMS. A window opens in which you can view items and select items to delete.
5. When you are done, click **Exit** to close the **DMS Helper.**

Issuing SQL Commands

You can use SQL Minus to issue SQL commands to the database.



TC SQL Minus

Driver Class Name: oracle.jdbc.driver.OracleDriver

Database URL: jdbc:oracle:thin:@DB_HOST:1521:DB_NAME

User Name:

Password:

Method: Issue a SQL command

1. Click **File > SQL-Minus**, or press **Ctrl** and the minus key (-).
2. In the **SQL Minus** dialog, select or enter the following SQL values:
   - **Driver Class Name**
   - **Database URL**
   - **User Name** and **Password**

Note: When you enter a value, it will be added to the drop-down list for future use.

3. To connect to the SQL database, click **Connect**.
4. To load an existing SQL script:
   a. Click **Load SQL**.
   b. Navigate to the script you want to load, then double-click it.
5. To run the command, click **Execute.** To cancel the command, click **Cancel.**

Tip: If any text in the edit pane is highlighted, only that highlighted text is executed when you click **Execute**.

6. To view information about the database or create a `SELECT` statement:
   a. Click **DB Info**. An explorer window of the database opens.
   b. To create a `SELECT` statement right-click one or more columns or tables, then click **Select**.

      A `SELECT` statement is inserted into the **SQL Minus** dialog.
7. When you are done, click **Exit** to close the **SQL Minus** dialog

Warning: If you are using Oracle, use `CREATE` and `DROP` statements carefully because they can cause data loss.

Chapter 10

Installing IStream Publisher

This chapter explains how to install IStream Publisher and the IStream Publisher Console.

This chapter describes:

- *The Installation Process* on page 100
- *Installing IStream Publisher* on page 101
- *Configuring WAS (Optional)* on page 121
- *The Directory Structure* on page 134

The Installation Process

The installation process consists of various stages and steps, some of which apply only in certain situations. For example, some stages or steps apply only to upgrades while others apply only to WebSphere Application Server configurations.

Note: For information on deploying Console and Web Services on other application servers, please see the *Release Notes*.

Upgrading IStream Publisher

If you are *upgrading* IStream Publisher, please note that:

- you can upgrade directly only from IStream Publisher 3.2 or 4.1
- the new version is installed in a new folder
- the database schema and the data in the Console database are migrated to the newer version
- the database schema and the data in the domain database are migrated to the newer version

Installing IStream Publisher

Installing IStream Publisher consists of the following stages:

- *Stage 1: Prepare the Installation* on page 102
- *Stage 2: Plan and Create a Queue Configuration* on page 107
- *Stage 3: Obtain Information* on page 109
- *Stage 4: Configure the Database* on page 113
- *Stage 5: Install the Rendering Services Printer Drivers* on page 115
- *Stage 6: Install Calligo or IStream Document Manager Components* on page 118
- *Stage 7: Run the Installation Application* on page 120

Note: If you are deploying IStream Publisher onto IBM WebSphere Application Server (WAS), you also need to complete the following stages:

- *Stage 8: Deploy IStream Publisher Console on WAS* on page 121
- *Stage 9: Deploy the Web Services Interface Application onto WAS* on page 129
- *Stage 10: Configure IStream Publisher for the OpenText DMS (Optional)* on page 131

Finally, you need to complete the installation:

- *Stage 11: Complete the Installation* on page 132

Stage 1: Prepare the Installation

Important: Before you can install IStream Publisher, you need to complete all the procedures described in this stage.

Step A: Check Your System Requirements

IStream Publisher is a distributed application. IStream Publisher agents and the IStream Publisher Console can be installed on one or more computers.

You will need separate systems for the supporting applications including the database server, the messaging service, and the application server.These are described in *Software Requirements* on page 104.

Note: For additional information about the IStream Publisher architecture, see the *IStream Publisher Architecture* chapter of the *IStream Publisher Administrator's Reference Guide.*

Creating an IStream Publisher Domain

When installing IStream Publisher, you will need to create an IStream Publisher domain.

Using OpenJMS

If you are using OpenJMS instead of IBM WebSphere MQ, you will also need to install the **OpenJMS Agent,** to provide the JMS messaging system.

Hardware Requirements

Your hardware requirements will vary greatly depending on your output requirements and the number of queues and services that will be implemented in your organization. However, at a minimum, ensure that the various systems which you will install IStream Publisher on meet the following hardware requirements:

Component	Minimum	Recommended
Console:		
Processor	1 GHz	2.8 GHz – single CPU or faster
RAM	512 MB	1 GB or higher
Free Disk Space	180 MB plus working storage (spooling, logs, temporary files) and so on	2 GB or greater on all drives and partitions plus log space
Worker:		
Processor	2 GHz	3.8 GHz – single CPU or faster
RAM	2 GB	2 GB
Free Disk Space	30-50 MB plus working storage (spooling, logs, temporary files) and so on	5 GB or greater on all drives and partitions plus log space 10,000 RPM hard drive
Author Client and End User Workstation:		
Processor	500 MHz	2 GHz – single or faster
RAM	128 MB	512 MB
Free Disk Space	2 GB plus disk space for documents created	5 GB
DMS Server:		
Processor	2 GHz	3.8 GHz – dual or faster
RAM	2 GB	4 GB
Free Disk Space	4 GB for the application, plus disk space for documents	4 GB for the application, plus disk space for documents and disk space for index
Assembly Server:		
Processor	3.2 GHz	3.8 GHz – single or faster
RAM	1.2 GB	2 GB
Free Disk Space	100 MB plus disk space for model documents to be stored locally	500MB for application and logs

Software Requirements

Please see the *IStream Publisher Release Notes* for the supported software versions of the following items:

- **Operating Systems** – Microsoft Windows
- **Databases and JDBC drivers**– for storing IStream Publisher and console data
- **JMS Messaging Services:** the JMS server type that manages the queue traffic
    - If you want to use **IBM WebSphere MQ**, it must already be installed on your system before installing IStream Publisher.
    - If you want to use **OpenJMS** instead of IBM WebSphere MQ, it will be installed during the installation of IStream Publisher.
- **Servlet Containers –** required for the IStream Publisher Console
- **IStream Document Manager** – for the Worker, and optionally for the DMS
- a **common repository** for storing files: this can be a shared file system, an FTP site or a DMS
- a print, email or fax server (if required to meet your business needs)
- **Microsoft Word**
- **Internet Explorer**

Note: If you are installing IStream Publisher on a Windows 2003 system, ensure the security template that you apply can run IStream Publisher agents. (For more information about security templates, go to the Microsoft TechNet website: http://technet.microsoft.com)

Deploying a Test Domain

For a test domain environment with a basic queue configuration, you will need at least one system containing a Worker.

You administer the IStream Publisher domain itself using the IStream Publisher Console.

Deploying a Production Domain

The number of Workers you need depends on:

- the expected volume, complexity and size of your documents
- the rendering services and delivery channels required
- the expected turnaround time of requests if used by a client in an interactive manner
- your infrastructure (network and servers)

Note: You cannot install Workers on the Solaris platform.

Step B: Plan Your Domain

Note: You only have to complete this procedure once for your entire domain.

1. Decide on a domain name for your deployment. You will need this name when you install an IStream Publisher agent.

   The domain name, its type and the host name of the system which the agent is running on make up an agent's JMX (Java Management Extensions) name. The JMX name uniquely identifies the system for administration.

Note: The host name cannot start with a number. The domain name is case-sensitive.

2. To administer multiple environments using IStream Publisher Console, each IStream Publisher environment must have a unique domain name using the same the Console database.
3. Ensure you enter the domain name consistently during all installations.
4. The IStream Publisher Console runs on IBM WebSphere, BEA WebLogic, or Tomcat. The Console supports the administration of one or more IStream Publisher domains, depending on your needs.

Step C: Assign User Rights

You need to assign specific access rights to the user who will be installing and run IStream Publisher agents as Windows services.

Method: Assign user rights

1. From the **Control Panel**, select **Administrative Tools** > **Local Security Policy > User Rights Assignments**.
2. Double-click **User Rights Assignments**, then double-click **Act as part of the operating system**.
3. Click **Add User or Group** to add the user to this group.
4. Click **OK** to save the changes, then restart your computer.

Step D: Uninstall IStream Publisher (Upgrades Only)

If you are upgrading from IStream Publisher, you will need to uninstall it first.

The uninstall program will remove all files except those of third-party applications such Oracle, IBM WebSphere MQ or Microsoft SQL Server.

Log files, config files and database scripts will be backed up in the `\backups` directory.

1. Ensure that:
   - the previous version of IStream Publisher is installed and working
   - you have applied all the required IStream Publisher patches
   - all work in progress and submitted requests are fully processed

2. Make a note of your existing database connection and queue information. You will need this information later when you reinstall.
3. Back up your Console and domain databases.
4. Stop all IStream Publisher Windows services and processes.
5. If you need to move the database, use the **Backup/Restore Database** feature in Microsoft SQL, instead of the **Export/Import Database** feature to ensure all the column values can be restored.
6. Record the database server name and credentials.
7. If you are using the Open JMS Server, stop it.
8. Close all files from the folder where IStream Publisher is installed.
9. Close all applications.
10. Run `uninstallServices.bat` located in your *`IStream Publisher install`*`\bin` folder.
11. Manually delete the IStream Publisher install directory.

IStream Publisher is now uninstalled from your system. Continue on to the next stage.

Stage 2: Plan and Create a Queue Configuration

Note: This stage applies only if you are using IBM WebSphere MQ. If you are using OpenJMS, go to *Stage 3: Obtain Information* on page 109.

This guide uses a simple queue configuration.

1. **Choose an IStream Publisher Configuration**

   There are two different configuration types for IStream Publisher:

   - **a basic queue configuration**
   - **a complex distributed queue configuration**

   The main differences between these configurations are:

   - the messaging dynamics
   - the volumes of messages that can be processed

   In high volume, performance-critical situations, a distributed queue deployment is preferred, but is a more complex system.

   The following step describes a *basic* queue configuration. If you require the complex distributed queue configuration, please contact Customer Support.

2. **Configure a Basic System**

   This step describes how to create a basic queue system deployment for a simple development or testing environment:

   Create a queue manager on your IBM WebSphere MQ server and the queues listed below. The physical queue names are examples only. You can use different names as long as they are consistent within your deployment.

Logical Queue	Physical Queue Name	Function
Submission Queue	submit	The queue where the client will submit all distribution, aggregate and simple requests.
Simple Service Queue & Distribution Service Queue	service	The queues where all Service Managers will pick up service requests. Also in this simple deployment we will be using this same queue to service distribution requests.
Completion Queue	completion	The queue to which the Loopback service sends the completion message to be picked up by the distribution service at the end of processing of a distribution request.
Response Queue	response	The queue where the client will monitor for responses from Service Managers when service requests are processed.
Admin Queue	admin	The queue where all administration messages are sent. Optional.

3. Record the information about the Queue Manager: the host name, port number, and queue manager name. You will need this information during the installation. Enter the information carefully. Note that all names are case-sensitive.

Note: The host name cannot start with a number.

4. Ensure that the queues:
   - use persistent mode
   - can allow a maximum of 640,000 messages
   - use a default priority level of 4
   - have a large enough **Maximum Message Size**
   - are the same ones you will specify during your IStream Publisher installation. You need to create a `JAVA.CHANNEL` for the Queue Manager on IBM WebSphere MQ server using **Server Connection Channel** in the **Advanced Settings** of the Queue Manager.
5. Ensure that you have enough available channels to support your IStream Publisher environment.
6. Ensure that Queue Manager’s log properties are large enough for your estimated number of requests. You can define these settings when installing the Queue Manager, and can change them using the **WebSphere MQ Services** console.

The recommended settings are:

- LogFilePages: 1024
- LogPrimaryFiles: 30
- LogSecondaryFiles: 20

Stage 3: Obtain Information

This section lists all the information that you will need to enter during the IStream Publisher installation.

Note: If you are upgrading, you will need to enter the same setup information that you used previously.

Step A: Obtain General Installation Information

JMS Server Type

- if you want to use **IBM WebSphere MQ Server**, it must first be installed on your system
- if you want to use a new **OpenJMS Server,** it will be installed during the installation of IStream Publisher

Database Information

- **Database type:** Microsoft SQL Server *or* Oracle

Admin Console Database Settings

You will need the following information to connect to the IStream Publisher Console database:

- **Host Name** – the host name of the system containing the Console database. This cannot start with a number.
- **Port** – the port number of the system containing the Console database. The default ports are 1433 for Microsoft SQL and 1521 for Oracle.
- **Database Instance Name** – the name of the Console database
- **User ID** and **Password** – the ID and password of a user who has read and write access to the Console database

IStream Publisher Database Settings

You will need the following information to connect to the IStream Publisher database:

- **Host Name** – the host name of the system containing the IStream Publisher database. This cannot start with a number.
- **Port** – the port number of the system containing the IStream Publisher database. The default ports are 1433 for Microsoft SQL server and 1521 for Oracle.
- **Database Instance Name** – the name of the IStream Publisher database
- **User ID** and **Password** – the ID and password of a user who has read and write access to the IStream Publisher database

IStream Publisher Domain Information

- **Domain Name** – the name that identifies the IStream Publisher domain to which this agent belongs. This should be the same name you chose in *Step B: Plan Your Domain* on page 105. It should also be the same for all IStream Publisher components that will make up a logical group to be administered together.
- **Agent Host Name** – the host name of the system where the selected Agents will be installed. It is used to distinguish between components in the IStream Publisher instance, and is usually the host name of the current system. This cannot start with a number.

Tip: To simplify your setup, make the **Agent Host Name** the same as your system name.

IStream Publisher Agents Services

- The **Username** and **Password** for the IStream Publisher Agents Services.

Note: You will need to enter the **Username** in this format:

`<host name>\<user name>`, for example, `IStream PublisherHost\jsmith`

Step B: Obtain Domain Information

For new installations, you will need to create a new domain. Gather the following information before installing. You will be prompted for it after the installation.

Note: An asterix (*) indicates a required field.

General Domain Information

- *** Domain Name** – the name of your IStream Publisher domain. For new installations, this must match the domain name that you specified during the installation.
- **Domain Description** – a description of the IStream Publisher domain
- *** JMS Server Type** – the type of JMS server: select **IBMWebSphere** or **OpenJMS**

IStream Publisher Database Information

- *** Database Server Host** – the name of the system hosting the IStream Publisher database. This cannot start with a number.
- *** Database Server Port** – the port number of the IStream Publisher database
- *** Database Name** – the name of the IStream Publisher database
- *** Username** – the username required to log into the IStream Publisher database

- *** Password** – the password required to log into the IStream Publisher database

You will enter the remaining information on the **Domain Settings** page. Note that certain fields appear depending on the **JMS Server Type** you selected.

OpenJMS Settings

This section appears only if you selected **OpenJMS** as the **JMS Server Type**

- *** OpenJMS Host Name** – the name of the system hosting OpenJMS. This cannot start with a number.

IBM WebSphere MQ Settings

This section appears only if you selected **IBM WebSphere MQ** as the **JMS Server Type.**

- *** MQ Host Name** – the name of the system hosting the Queue Manager. This cannot start with a number.
- *** MQ Port** – the port number of the system containing the Queue Manager
- *** Queue Manager Name** – the name of the Queue Manager
- **MQ Channel Name** – the channel name for the IBM WebSphere MQ Queue
- *** Admin Queue** – the name of the Admin Queue
- **Username** – the username required to log into the Queue Manager
- **Password** – the username required to log into the Queue Manager

Default QueueSets Settings

- *** Submission Queue Name** – the name of the submission queue
- *** Service Queue Name** – the name of the service queue
- Distribution Service Queue Name – the name of the distribution service queue

Distribution Service Settings

- *** Completion Queue Name** – *for IBM WebSphere MQ only:* the name of the Queue used to process completion requests from the distribution service. For simple domains, this name should be the service queue name
- *** Shared Location URL** – The URL pointing to the common file repository. All Worker systems use this field to store intermediate temporary files during distribution. All Worker systems must be able to access this location.

If you are using a file share connection, enter `file://server/share/` to connect.

If you are using an FTP connection, enter `ftp://serverurl` to connect.

- **Shared Location Username** – the username required to log into the **Shared Location URL**
- Shared Location Password – the password required to log into the **Shared Location URL**

Delivery Service Settings

- **SMTP Host (fax/email)** – the hostname of the SMTP server used for fax and email delivery services
- **Fax Gate Mail Box** – the SMTP mail account to which fax requests from IStream Publisher will be sent. You need to configure RightFax's SMTP gateway service to use this account to receive IStream Publisher fax service requests.
- **Fax From Address** – a valid email account that will receive notifications from RightFax

Worker Settings

- *** Temp Folder** – the folder that Workers will use to store local temporary files, for example, `C:\Temp`

You are now ready to install or upgrade IStream Publisher.

Stage 4: Configure the Database

Step A: Install the Required Drivers

The required database drivers are included with IStream Publisher. However, if you are using IBM WebSphere MQ, you will need to copy certain `.jar` files in your IBM WebSphere MQ server `\java\lib` folder to a central location that all agents can access. You will need to provide this location during the installation.

The `.jar` files that you need to copy depend on

- the version of IBM WebSphere MQ you are using

  -and-

- whether you are also using WAS, as follows:

If you are using IBM WebSphere MQ 5.3.x, the required files are:

- `com.ibm.mq.jar`
- `com.ibm.mqjms.jar`
- `jms.jar`
- `connector.jar`

If you are using IBM WebSphere MQ 6.0.x *without* the IBM WebSphere Application Server (WAS), the required files are:

- `com.ibm.mq.jar`
- `com.ibm.mqjms.jar`
- `jms.jar`
- `connector.jar`
- `dhbcore.jar`
- `jta.jar`

If you are using IBM WebSphere MQ 6.0x *with* the IBM WebSphere Application Server, the required files are:

- `com.ibm.mq.jar`
- `com.ibm.mqjms.jar`
- `dhbcore.jar`
- `jta.jar`

Step B: Create the Databases

There are two databases that IStream Publisher uses:

- **Domain Database** – This database stores request log message, temporary records of Distribution Request, OpenJMS tables (when they are used), and other information, but does not store any configuration information.

The database only contains information that pertains only to a single IStream Publisher domain.

- **Console Database** – This database stores configuration information such as IP addresses, ports, URLs for temporary directories, the system config XML file, and so on. This database can store information about multiple domains.

Method: Using Single or Multiple Physical Databases

The Domain Database and the Console Database are logical databases. These databases can be in the same physical database or in separate physical databases.

If you are using the IStream Publisher Console to administer a *single* IStream Publisher domain, you could use a single physical database for the Domain and Console databases. However, we recommend using two different databases so that you can back them up separately.

If you are using the IStream Publisher Console to administer *multiple* IStream Publisher domains, then you should create the Domain and Console databases in separate physical databases.

1. On your SQL or Oracle database, create your Console database and User for IStream Publisher Console. Note the host name, port, database name (or instance name), username and password.
2. Create a second database and user for the IStream Publisher agents and again note the host name, port, database name (or instance name) and username and password.

Note: Host names cannot begin with a number.

The users for both databases should be the owner of each and have full permissions for modifying, deleting and adding data. You will need this information for both the IStream Publisher and IStream Publisher console installation.

The Console database does not contain a large amount of data, but it contains configuration data that you back up regularly. The IStream Publisher database is used dynamically to store temporary information as well as auditing information, so it usually grows over time.

Stage 5: Install the Rendering Services Printer Drivers

Before you install the Worker, and depending on the services that will run on the Worker, you need to first install the printer drivers that are required for the rendering services.

The installation process checks if these drivers are installed. If they are missing, an error message appears.

Step A: Configuring PCL Rendering Services

Complete the following procedure to create the virtual driver for each Worker that renders to PCL format.

Required software: Microsoft Word and a PCL printer driver

Method: Install the printer driver for PCL rendering services

1. On the Worker system, click **Start > Settings > Printers > Add Printer**.
2. In the first window of the **Add Printer** wizard, click **Next**.
3. In the next window, choose **Local Printer**.
4. Clear the option to have Windows detect the printer, then click **Next**.
5. Select **Create a new port**, create a new **Local Port**, then click **Next**.
6. In the **Create Port** window, name the port **PCL**, then click **Next**.
7. Select the PCL print driver**,** then click **Next**.
8. When asked if you want to keep the existing driver, select **Yes**, then click **Next**.
9. When asked if you want to share the printer, select **No**, then click **Next**.
10. When asked if you want to print a test page, select **No**, then click **Next**.
11. On the summary window, click **Finish**.

Step B: Configuring Postscript Rendering Services

Complete the following procedure to create the virtual driver for each Worker running the PS (PostScript) rendering services.

Required software: Microsoft Word and HP Postscript virtual driver

Method: Install the printer driver for PostScript rendering services

1. On the Worker system, click **Start > Settings > Printers > Add Printer**.
2. In the first window of the **Add Printer** wizard, click **Next**.

3. In the next window, choose **Local Printer**.
4. Clear the option to have Windows detect the printer, then click **Next**.
5. Select **Create a new port**, create a new **Local Port**, then click **Next**.
6. In the **Create Port** window, enter **PS** for the driver name, then click **Next**.
7. In the next window, select the PostScript printer driver, then click **Next**.
8. When asked if you want to keep the existing driver, select **Yes**, then click **Next**.
9. In the next window, ensure that the printer is not selected as the default driver, then click **Next**.
10. When asked if you want to share the printer, select **No**, then click **Next**.
11. When asked if you want to print a test page, select **No**, then click **Next**
12. On the summary window, click **Finish**.

Step C: Configuring Word to TIFF or PDF Rendering Service

Note: You must have the appropriate Publisher license to run this service. Please check your software license or contact Customer Support to determine if you are licensed to run this service.

Required software: Microsoft Word, Amyuni PDF Converter

The Amyuni PDF Converter is installed with the components on the Worker, as described in *Stage 6: Install Calligo or IStream Document Manager Components* on page 118.

Step D: Running the PDF to PCL Rendering Service

Note: You must have the appropriate Publisher license to run this service. Please check your software license or contact Customer Support to determine if you are licensed to run this service.

Step E: Running Other Services

Method: Run the Word to HTML rendering service on the Worker

To run the Word to HTML rendering service on the Worker, Microsoft Word must be installed on the Worker.

Method: Run the PDF to PS rendering service on the Worker

Note: You must have the appropriate Publisher license to run this service. Please check your software license or contact Customer Support to determine if you are licensed to run this service.

Method: Run the CLG to Word rendering service on the Worker

Ensure the required components are installed: see *Install Calligo or IStream Document Manager Components* on page 118.

Stage 6: Install Calligo or IStream Document Manager Components

This section describes how to install Calligo Enterprise 5.4 or IStream Document Manager 6.x components on each system that will be running as a Worker and providing Content Services.

Note: *Calligo Enterprise* is the previous name for *IStream Document Manager*. The IStream Document Manager 6.x application and manuals are included in the installation package. See the *IStream Document Manager Installation Guide* or the *Calligo Enterprise Manager Installation Guide* for detailed instructions on configuring and testing the installed components.

Method: Install Calligo Enterprise or IStream Document Manager components on the worker

1. Close all programs that are running.

2. If you have installed Assembly Server, ensure the Assembly Server service is stopped and the startup type is set to **Manual.**

Warning: **Do not** reboot the Worker system when prompted to after the component installation. Only reboot after IStream Publisher is fully installed.

3. Run the setup application as follows:

- **For Calligo Enterprise 5.4**: Run the Calligo `setup.exe` application from your current Calligo Enterprise 5.4 installation package.
- **For IStream Document Manager 6.x**: On the Worker systems where IStream Document Manager 6.x will be installed, run `setup.exe` from the **IStream Document Manager** folder on the IStream Publisher installation package.

Important: Only Calligo Enterprise 5.4 or IStream Document Manager 6.x should be used with IStream Publisher. These versions contains specific functionality necessary for IStream Publisher. **Do not install any other versions on the Worker.**

4. Click **Next** and complete the required values in each window.

5. On the **Select Features** window:

- **For Calligo 5.4:** Ensure that the **Workstation > Author** and **XML InfoSource** features are selected.
- **For IStream Document Manager 6.x:** Ensure that the **Workstation > Author** feature is selected.

6. If this is a new installation of Calligo or IStream Document Manager, the **Calligo** or **IStream DMS InfoSources** window will appear.

If you have a Calligo or **IStream DMS**:

- Enter the **Server**, **Port** and **Connection**, then click **Next.**

If you do *not* have a Calligo or **IStream DMS**:

- Click **Next** on each window that appears, and accept all the default values.

7. Continue to click **Next** and follow the onscreen instructions.

8. When the installation is complete, click **Finish**.

The following sections describe the process of running the actual installation application to install or upgrade IStream Publisher

Stage 7: Run the Installation Application

1. Ensure that you have administrative rights for the system you will be installing IStream Publisher on, otherwise you will not be able to install it.
2. Close all running programs.
3. From the root of the installation package, run `install.bat`.
4. Follow the installation wizard to complete the installation process.

Upgrading Information

If you are upgrading, please note that:

- If you are using an OpenJMS server, on the **JMS Server Type** dialog, select **Install Open JMS Server** and *not* **Connect to existing OpenJMS Server**.
- You should upgrade all agents that are in the same domain together.
- You must use the same setup information as in the previous installation.

Deploying Tomcat

If you want to deploy IStream Publisher under Tomcat, select the **Deploy IStream Publisher Console** check box in the **Optional Components** dialog during the installation.

Verifying the Database

The installation process verifies your database setting by checking the database connection. If the connection fails, you can ignore and continue the installation or you can go back and change the database settings.

If you selected to deploy IStream Publisher or Web Services, you can change the database setting by editing the following file, after the installation:

```
IStream Publisher_install_folder/tomcat/conf/Catalina/
localhost/rmc.xml
```

Note: If you are deploying the IStream Publisher Console onto the WebSphere Application Server (WAS), continue to the next section: *Configuring WAS (Optional)* on page 121. Otherwise, go directly to *Stage 11: Complete the Installation* on page 132.

Configuring WAS (Optional)

Important: The following two stages apply only if you are deploying the IStream Publisher Console onto the WebSphere Application Server (WAS):

- *Stage 8: Deploy IStream Publisher Console on WAS* on page 121
- *Stage 9: Deploy the Web Services Interface Application onto WAS* on page 129

Note: If you are using WebLogic, please consult your WebLogic documentation.

If you are *not* using WAS, please continue directly to *Stage 11: Complete the Installation* on page 132.

Stage 8: Deploy IStream Publisher Console on WAS

IStream Publisher Console is deployed using the standard WebSphere deployment procedure. This section provides an overview of this procedure. See your WebSphere documentation for details.

Note: Throughout this stage, note that:

- **for the IBM WebSphere Base edition:** the node name is the server host name: this cannot start with a number
- **for the IBM WebSphere Network Deployment and Enterprise editions:** the node name is the host name of the server where you will install IStream Publisher: this cannot start with a number

Step A: Confirm the Server Is Running

For Windows:

- In **Control Panel** > **Services**, confirm that the IBM WebSphere Application Server and IBM HTTP Server services are running.

For Unix:

- Enter the following command to check if the IBM WebSphere Application Server and IBM HTTP Server processes are running.

  ```
  #ps -x
  ```

  If they are not running, start them by entering the following commands:

  - **for the IBM WebSphere Application Server, enter:**

    ```
    #cd /opt/WebSphere/AppServer/bin
    #./startServer.sh server1
    ```

  - **for the IBM HTTP Server:**

    ```
    #cd /opt/IBMHttpServer/bin
    #./apachectl start &
    ```

Step B: Create a JDBC Provider

In this step, you open the WebSphere Administrative Console and then create and configure a JDBC provider.

1. **For Windows:**

   - Click **Start** > **Programs** > **IBM WebSphere** > **Application Server** > **Administrative Console.**

   For Unix:

   - In your web browser, go to `http://`*`host_name`*`:9090/admin`

2. If required, enter a **username** and **password**, then click **OK.**

   The WebSphere Administrative Console opens.

3. From the navigation tree on the left side, expand **Resources**.
4. Click **JDBC Providers**.
5. In the **Scope** section, click **Node**, then **Apply**.
6. Click **New** to create a new JDBC provider.

   The **JDBC Providers Configuration** page opens.

7. In the **JDBC Provider** drop-down list, select the type of provider to create depending on your current configuration:

   - **for Microsoft SQL server**: select **User-defined JDBC Provider**
   - **for Oracle**: select **Oracle JDBC Driver**

Warning: Do not select the (XA) version drivers because they do not work with IStream Publisher.

8. Click **Apply**.

   A second **JDBC Providers Configuration** page opens.

9. Enter the JDBC provider properties:

   - **Name** – a name for the provider: enter a descriptive name, for example, `IStream PublisherJDBCProvider`
   - **Description** – a description of the provider
   - **Classpath** – a list of paths to the required database driver files, for example:
     - **for Microsoft SQL**:

       `<driver path>\sqljdbc.jar`
     - **for Oracle 9i or 10g**

       `<driver path>\ojdbc14.jar`

       `<driver path>\orai18n.jar`

Note: These driver files are located at:

`\`*`IStream Publisher_install_folder`*`\drivers\jdbc`

Separate each classpath with the **Enter** key.

- **Implementation Classname** – the name of the Java data source class that is used to connect to the database: this name is provided for you when you select the JDBC provider type, which is:

  For MSSQL:

```
com.microsoft.sqlserver.jdbc.SQLServerConnectionPoolDataSource
```

  For Oracle:

```
oracle.jdbc.pool.OracleConnectionPoolDataSource
```

This class must be available in the driver file that the **Classpath** specifies.

10. Click **OK**.

11. Click the **Save** link near the top of the page, then click the **Save** button.

Step C: Create the JDBC Provider Data Source

After creating and configuring the JDBC provider, you can then create a data source under it. The data source will use the JDBC provider classes to connect to the database.

1. From the navigation tree on the left side, expand **Resources**.

2. Click **JDBC Providers**.

3. In the **Scope** section, click **Node**, then **Apply**.

4. Click the JDBC provider that you created for IStream Publisher in step 9 on page 122, for example, `IStream PublisherJDBCProvider`.

The configuration window for this JDBC provider opens.

5. Scroll down to the **Additional Properties** table

6. Select **Data Sources**.

Any data sources defined for the selected scope are listed.

7. Click **New** to create a new data source.

8. Complete the following fields for the new data source:

- **Name** – the name by which to administer the data source, for example, `IStream PublisherDataSource`
- **JNDI Name** – the name of the data source as registered in the application server’s name space, for example, `jdbc/Console`
- **Description** – a description of the data source
- **Category** – a category that you can use to classify the data source

9. Click **Apply**.

10. Scroll down to the **Related items** table, then select **J2C Authentication Data Entries**.

11. All **Authentication Data** defined for the selected DataSource are listed.

12. Click **New** to create new authentication data.

13. Complete the following fields for the new data source:
    - **Alias -** the name of the authentication data entry
    - **User ID** – the user name for connecting to the database: if the user ID is specified, the password must also be specified
    - **Default Password** – the password for connecting to the database: if the password is specified, the user ID must also be specified
14. Click **OK**.
15. Click the **Save** link near the top of the page, then click the **Save** button.
16. Repeat steps 1 to 6 to select the data source that you created.
17. Scroll down to the **Additional Properties** table, then select **Custom Properties**.

    The **Custom Properties** page opens. These properties consist of **Names** and their corresponding **Values**.

    for MSSQL:

    - click **New**, then enter the following values:
        - **Name** – `URL`
        - **Value** – `jdbc:sqlserver://`*`hostname`*`:`*`port number`*`;` `SelectMethod=cursor;DatabaseName=`*`databasename`*

    for Oracle:

    - enter the following **Value** of the **URL** for the Oracle thin driver provider:

      `jdbc:oracle:thin:@`*`hostname`*`:`*`port number`*`:`*`databasename`*
    - **driverType -** `thin`
18. Click **OK**.
19. Click the **Save** link near the top of the page, then click the **Save** button.
20. Repeat steps 1 to 6 to select data source that you created.
21. Scroll down to the **General Properties** table, and ensure the default value in **Datasource Helper Classname** field is:

    for MSSQL:

    `com.ibm.websphere.rsadapter.GenericDataStoreHelper`

    for Oracle:

    `com.ibm.websphere.rsadapter.OracleDataStoreHelper`
22. From the **Container-managed Authentication Alias** drop-down list, select the authentication data created in step 13.
23. From the **Mapping-Configuration Alias** drop-down list, select `DefaultPrincipalMapping`.
24. Click **OK**.

25. Click the **Save** link near the top of the page, then click the **Save** button.

Step D: Test the JDBC Provider Data Source

In this step, you test the JDBC provider data source for the database.

1. From the navigation tree on the left side, expand **Resources**.

2. Click **JDBC Providers**.

3. In the **Scope** section, click **Node**, then **Apply**.

4. Click the JDBC provider you created for IStream Publisher in step 9 on page 122, for example, `IStream PublisherJDBCProvider`.

5. Scroll down to the **Additional Properties** table.

6. Click **Data Sources**.

7. Select the check box beside the JDBC datasource name you entered in step 8 on page 123.

8. Click **Test Connection**.

9. If the test passes, continue to *Step E: Create the JMS Library* on page 125.

If the test fails, repeat the following steps and ensure that the settings within these steps are correct:

- *Step B: Create a JDBC Provider* on page 122
- *Step C: Create the JDBC Provider Data Source* on page 123

Step E: Create the JMS Library

In this step, you define a shared library for MQ Series.

Note: Complete this step only if you are deploying WAS with IBM MQSeries.

1. From the navigation tree on the left side, expand **Environment**.

2. Click **Shared Libraries**.

3. In the **Scope** section of the shared library definition, click **Node**, then **Apply**.

4. Click **New**.

5. Enter the **Name** for the **JMSLibrary** properties. This is the shared library name, for example, `MQLibrary`

6. Enter the **Classpath**(s) for the **JMSLibrary** that points to the `.jar` files in the IBM MQ Series classes directory

For MQ Series 6.0.x with WAS, the classpath needs to point to:

- `<driver path>\com.ibm.mq.jar`
- `<driver path>\com.ibm.mqjms.jar`
- `<driver path>\dhbcore.jar`
- `<driver path>\jta.jar`

For MQ Series 5.0.x with WAS, the classpath needs to point to:

- `<driver path>\com.ibm.mq.jar`
- `<driver path>\com.ibm.mqjms.jar`
- `<driver path>\jms.jar`
- `<driver path>\connector.jar`

7. Scroll down, then click **OK**.
8. Click the **Save** link, then click the **Save** button.

Step F: Deploy the IStream Publisher Console on WAS

In this step, you install IStream Publisher Console on the WebSphere Application Server by deploying the IStream Publisher Console WAR file.

1. From the navigation tree on the left side, expand **Applications**.
2. Click **Install New Application**.

   The **Preparing for the application installation** page opens.
3. Browse to the following file on the installation package:

   `\console\IStreamPublisherConsole.war`
4. Select the WAR file, then click **Open**.
5. Scroll down the page.
6. In the **Context Root** field, enter `rmc`, then click **Next**.

   A second **Preparing for the application installation** page opens.
7. Click **Next.**
8. If an **Application Security Warning** page appears, you can disregard it and click **Continue**.

   The **Step 1: Provide options to perform the installation** page opens.
9. Ensure that **Select pre-compile JSP is not checked**.
10. Scroll down then click **Next**.

    The **Step 2: Map resource references to resources** page opens.
11. Each resource reference defined in the application must be mapped to the corresponding resource.

    For WAS Base Edition:

    a. From the **Resource JNDI name** drop-down list, select `<`*`servername`*`>:jdbc/Console`

    b. Select the checkbox next to the **Module** column heading on the left side of the table.

    c. Click **Apply**.

    The **JNDI Name** for each module should automatically be updated to `jdbc/Console`.

d. Click **OK** and continue to the next step.

For WAS Network Deployment or Enterprise Editions – for the **Resource JNDI name**, select the IStream Publisher installed server node name, for example: `<servernodename>:jdbc/Console`

12. Scroll down then click **Next**.

The **Step 3: Map virtual hosts for Web modules** opens.

13. Click **Next**.

The **Step 4: Map modules to application servers page** opens.

14. Select the server which each module should be deployed on. IBM recommends that all modules from one application be deployed onto a single server.

- **for WAS Base Edition** – make no changes: continue to step 15.
- **for WAS Network Deployment or Enterprise Editions**:

a. From the **Clusters and Servers** drop down list, select the IStream Publisher installed server.

b. Select the check boxes for all the **Module**s.

c. Click **Apply**.

15. Click **Next**.

The **Step 5: Summary** page opens

16. The **Summary** page lists the application deployment settings. Review these settings, scroll down, then click **Finish**.

17. Click **Save to Master Configuration**, then click **Save**.

Step G: Add a Environmental Entry

Note: Complete this step only if:

- you are deploying WAS 5.1.x or earlier with Oracle

-OR-

- WAS is using IBM SDK v1.4.2 or earlier

1. From the navigation tree on the left side, expand **Servers**.

2. Click **Application Servers**.

3. Select `server1` as your application server.

4. Under **Additional Properties**, click **Process Definition**.

5. Again under **Additional Properties**, click **Environment Entries**.

6. Click **New** to create a new environment entry.

7. Name the entry `JITC_COMPILEOPT`.

8. Assign the following **Value** for the entry:

```
SKIP{oracle/jdbc/driver/T4CPreparedStatement}{doOall8}
```

9. Click **OK**.
10. Click the **Save** link near the top of the page, then click the **Save** button.

Step H: Change Application Properties and Add the Libraries

In this step, you will change the application properties and add the required libraries to the application.

1. From the navigation tree on the left side, expand **Applications**.
2. Click **Enterprise Applications**.
3. Click the **IStreamIStream PublisherConsole_war** link.
4. Scroll down to the **General Properties**.
5. Select the **Use Metadata From Binaries** check box.
6. From the **Classloader Mode** drop-down list, select `PARENT_LAST`.
7. From the **WAR Classloader Policy** drop-down list, select `Application`.
8. Click **Apply**.
9. If you are deploying WAS with MQ Series, complete these steps:
   a. Scroll down to the **Additional Properties** section.
   b. Click the **Libraries** link.
   c. Click **Add**. The library you created opens.
   d. In the **Library Name** column, click the JMS library name that you created in *Step E: Create the JMS Library* on page 125.
10. Click **OK**.
11. Click the **Save** link near the top of the page, then click the **Save** button.

Stage 9: Deploy the Web Services Interface Application onto WAS

In this stage, you use the WebSphere Administrative Console to deploy the Web Services Interface Application onto the application server.

Step A: Deploy the IStream Publisher Console

If you did not already complete *Stage 8: Deploy IStream Publisher Console on WAS* on page 121, then complete steps A to E in this stage.

Step B: Deploy the Web Services Interface WAR file

1. From the navigation tree on the left side, expand **Applications**.
2. Click **Install New Application**.

   The **Preparing for the application installation** page opens.
3. Browse to the following file on the installation package:

   `\web-services\IStreamPublisherWSI.war`
4. Select this file, then click **Open**.
5. Scroll down the page.
6. In the **Context Root** field, enter `wsi`, then click **Next**.

   A second **Preparing for the application installation** page opens.
7. Click **Next.**
8. If an **Application Security Warning** page appears, disregard it and click **Continue**.

   The **Step 1: Provide options to perform the installation** page opens.
9. Ensure that the **Select pre-compile JSP** check box is not selected.
10. Scroll down then click **Next**.

    The **Step 2: Map resource references to resources** page opens.
11. Each resource reference defined in the application must be mapped to the corresponding resource.

    For WAS Base Edition:

    a. From the **Resource JNDI name** drop-down list, select `<servername>:jdbc/Console`

    b. Select the checkbox next to the **Module** column heading on the left side of the table.

    c. Click **Apply**.

    The **JNDI Name** for each module should automatically be updated to `jdbc/Console`.

    d. Click **OK** and continue to the next step.

For WAS Network Deployment or Enterprise Editions – for the **Resource JNDI name**, select the IStream Publisher installed server node name, for example: `<servernodename>:jdbc/Console`

12. Scroll down then click **Next**.

The **Step 3: Map virtual hosts for Web modules** opens.

13. Click **Next**.

The **Step 4: Map modules to application servers page** opens.

14. Select the server which each module should be deployed on. IBM recommends that all modules from one application be deployed onto a single server.

- **for WAS Base Edition** – make no changes: continue to step 15.
- **for WAS Network Deployment or Enterprise Editions**:

a. From the **Clusters and Servers** drop down list, select the IStream Publisher installed server.

b. Select the check boxes for all the **Module**s.

c. Click **Apply**.

15. Click **Next**.

The **Step 5: Summary** page opens

16. The **Summary** page lists the application deployment settings. Review these settings, scroll down, then click **Finish**.

17. Click **Save to Master Configuration**, then click **Save**.

Step C: Add the Libraries

Note: Complete this step only if you are deploying WAS with IBM MQSeries.

In this step, you add the required libraries to the application.

1. From the navigation tree on the left side, expand **Applications**.

2. Click **Enterprise Applications**.

3. Click the **IStreamIStream PublisherWSI_war** link.

4. Scroll down to the **Additional Properties** section.

5. Click the **Libraries** link.

6. Click **Add**. The library you created opens.

7. In the **Library Name** column, click the JMS library name that you created in *Step E: Create the JMS Library* on page 125, then click **OK**.

8. Click the **Save** link, then click the **Save** button.

9. Ensure the queue server is started before starting the Worker.

You can now start the new applications in WebSphere.

Stage 10: Configure IStream Publisher for the OpenText DMS (Optional)

If you are using the OpenText DMS as the repository, complete this procedure. Otherwise, please continue to the next stage.

1. Copy `\lapi\lapi_CALLIGO.jar` from the installation package to the `\etc folder`, typically:

   `C:\Whitehill\IStream\Publisher-4.2\etc\`

2. Download `LAPI_970_WIN.exe` from the OpenText Livelink 9.7.0 SDK package, then run this file to install the package.
3. Copy `lapi.jar` under the `\lib` folder, typically:

   `C:\Program Files\Open Text\LAPI\lib\lapi.jar`

   to the IStream Publisher installed `\etc` folder, typically:

   `C:\Whitehill\IStream\Publisher-4.2\etc`

4. Continue to the next stage to complete the installation.

Stage 11: Complete the Installation

In this stage, you complete the IStream Publisher installation.

Step A: Start the IStream Publisher Web Server Service

Note: Complete this step only if you selected to deploy IStream Publisher on Tomcat.

1. If you chose to install the IStream Publisher Windows services, start the IStream Publisher Web server service from **Control Panel** > **Administrative Tools > Services**.
2. If you chose not to install the IStream Publisher Windows services, start the IStream Publisher Web server service manually by running `startWebServer.bat` file in the `\bin` folder of the IStream Publisher installation directory, typically `C:\Whitehill\IStream\Publisher-4.2\`

Step B: Add a Domain

If this is a new installation, you will need to add an IStream Publisher domain. For detailed steps, see *Adding a Domain* on page 66. For field descriptions, see *Obtain Domain Information* on page 110.

Step C: Start the Remaining IStream Publisher Windows Services

1. If you chose to install the IStream Publisher Windows services, start these from **Control Panel** > **Services**.
2. If you chose not to install the IStream Publisher Windows services, start these services manually by running the necessary `.bat` files located in the bin folder of the IStream Publisher installation directory, typically `C:\Whitehill\IStream\Publisher-4.2\`

Note: If you chose OpenJMS as the JMS server, start the **OpenJMSAgent** service first. Otherwise, you can start the IStream Publisher services in any order.

Step D: Configure Oracle

Complete the following procedure if:

- the Worker is installed as a Windows Service *and*
- the Assembly engine uses the Oracle ODBC

Method: Configure the Oracle client

1. Install the Oracle ODBC client (if required) using the user name and password used to start the **Worker** service.
2. From the **Control Panel**, click **System**, then select the **Advanced** tab and click **Environment Variables...**.
3. In the **User Variables...** window, select the **Path** variable.
4. Click **Edit**, select all of the text in the **Variable Value** text box, then press `CTRL-C` to copy this text.
5. In the **System Variables** section, double-click the **Path** variable.
6. Press `CTRL-V` to paste the copied text after the semi-colon into the **Variable Value**, then click **OK**.
7. Reboot the system.

Step E: Test the Connection

Use a simple rendering request, for example, rendering a Word to PDF to test the connection of your IStream Publisher System. You can use the Test Console as described in *The Test Console* on page 75.

The IStream Publisher installation is now complete.

Important: If you restart the OpenJMS server, the IBM WebSphere MQ server or your database server, you must restart all your IStream Publisher Agents.

The Directory Structure

The following folders are added to your system by the installation:

Default Path	Description

Windows:
`C:\Whitehill\IStream\IStream Publisher-4.2`
Unix:
`opt/Whitehill/IStream/ IStream Publisher-4.2`
	Main installation folder

install_folder\bin	Contains the Agents' batch files. If you chose to run IStream Publisher Agents as Windows Services, launch their respective batch launch files.
install_folder\drivers\jdbc	Contains the database drivers
install_folder\drivers\jms	Contains the IBM MQ drivers. Present only if MQ was chosen as the JMS Server.
install_folder\logs	Contains the worker log file which traces the worker's run time status, the generation log for generation requests, and the installation log file.
install_folder\dtd	Contains the system DTD and schema files.
install_folder\tomcat\webapps\rmc	Contains the IStream Publisher Console
install_folder\tomcat\webapps\wsi	Contains the Web Services Interface
install_folder\openjms	Contains the Open JMS server. Present only if OpenJMS was chosen as the JMS server.
install_folder\tomcat	Contains the Tomcat web server. Present only IStream Publisher Console was selected to deploy during the installation, instead of deploying on WAS

8.

Chapter 11

Uninstalling and Reinstalling

This chapter describes:

- *Uninstalling IStream Publisher* on page 136
- *Reinstalling IStream Publisher* on page 138

Uninstalling IStream Publisher

This section describes how to uninstall IStream Publisher from your system.

Note: When you uninstall IStream Publisher, the following items are not removed:

- the IStream Publisher folders and database files
- any third-party applications such as Oracle, IBM WebSphere MQ or Microsoft SQL Server

Important: Before uninstalling IStream Publisher, back up all IStream Publisher database files that you want to keep.

Step A: Undeploy IStream Publisher Console

Complete this step to undeploy IStream Publisher Console from the IBM WebSphere Application Server.

1. Undeploy IStream Publisher Console:

a. In WAS, from the navigation tree on the left side, expand **Applications**.

b. Click **Enterprise Applications**.

c. Select the **IStreamIStream PublisherConsole_war** check box, and then click **Stop**.

d. Select the **IStreamIStream PublisherConsole_war** check box, and then click **Uninstall**.

e. Click the **Save** link, then click the **Save** button.

2. Remove the JDBC Provider:

a. From the navigation tree on the left side, expand **Resources**.

b. Click **JDBC Providers**.

c. In the **Scope** section, click **Node**, and then **Apply**.

d. Click the JDBC provider you created for IStream Publisher Console in *Step B: Create a JDBC Provider* on page 122.

e. Select the check box next to this JDBC provider, then click **Remove**.

f. Click the **Save** link, then click the **Save** button.

3. Remove all shared libraries:

a. From the navigation tree on the left side, expand **Environment**.

b. Click **Shared Libraries**.

c. In the **Scope** section, click **Node**, and then **Apply**.

d. Select the check box next to the **IStream Publisher Console** library name that you created in *Step E: Create the JMS Library* on page 125, and then click **Remove**.

- **e.** Select the check box next to the JMS library name that you created for IStream Publisher Console in *Step E: Create the JMS Library* on page 125, then click **Remove**.
- **f.** Click the **Save** link, then click the **Save** button.

Step B: Uninstall IStream Publisher Services

1. Stop all IStream Publisher Windows Services.

2. Close all files from the folder where IStream Publisher is installed.

3. Close all applications.

4. For Windows, run `uninstallServices.bat` in the IStream Publisher `bin` folder, usually located at:

```
C:\Whitehill\IStream\Publisher-4.2\bin
```

Warning: **Ensure that you run this batch file before continuing to the next step!**

5. After running the batch file, you can delete the IStream Publisher installation folder.

IStream Publisher is now uninstalled from your system.

Reinstalling IStream Publisher

If you are reinstalling to add new components:

Keep the previous installation settings and complete the installation process to add the new components, but ensure that the database settings, the MQ or Open JMS settings and the domain name remain exactly the same as the previous installation.

If you want to uninstall specific components:

You cannot uninstall components by reinstalling IStream Publisher onto an existing installation and clearing the component check boxes during the reinstallation process. The only way to remove previous components is to uninstall IStream Publisher, and then reinstall it, selecting only the components you want.

If the previous installation is completely uninstalled: the reinstallation process will be the same as a brand new installation.

Chapter 12

Reports

This chapter describes:

- *IStream Publisher Reports* on page 140
- *Installing IStream Publisher Reports* on page 141
- *Using IStream Publisher Reports* on page 142

IStream Publisher Reports

IStream Publisher reports consist of four model documents, which each have two sections.

The reports are:

- TransactionSummaryReport.CMS – a high-level report of all processed requests
- TransactionDetailReport.CMS – a detailed report of all processed requests, grouped by request types
- DistributionDetailReport.CMS – details of a specific distribution request
- ErrorDetailReport.CMS – details of a specific error

A Report view is created when you create a new domain database, or if you have migrated the database from a previous release. This view is based on the Request and StatusOrder tables.

All model documents query the database using the Report view, except for ErrorDetailsReport.CMS, which queries the ErrorInfo table.

Installing IStream Publisher Reports

1. Copy the contents of the `\IStream Publisher\Reports` folder on the IStream Publisher installation package to a folder on your system.
2. Create a file system InfoSource named `IStream PublisherReportsFS` to hold the reports. Configure this InfoSource to point to the location of the model documents that you copied over in the previous step.
3. Create a `UserDatabase` InfoSource named `Pub41DS`. This InfoSource will retrieve data through an ODBC datasource configured to connect to the IStream Publisher domain database.

Using IStream Publisher Reports

1. Open IStream Author.
2. Select **Author** > **User Info** and enter the IStream Publisher domain database login user and password for the user proxy login information
3. Select **Author** > **Options** > **Repository**, and select `IStream PublisherReportsFS` as the default repository.
4. Open a model document report and set the view to **Print Layout**.
5. Click the generate button.
6. Enter the following report criteria, depending on the report you selected:
   - `TransactionSummaryReport.CMS` – Enter a start date, and optionally an end date: see *Entering Start and End Times* on page 142
   - `TransactionDetailReport.CMS` – Enter a start date and an end date (optional). For details, see *Entering Start and End Times* on page 142. You can also enter a filter (optional). This filter is a SQL expression based on the JMS headers.
   - `DistributionDetailReport.CMS` – Enter a specific request ID; request IDs appear in the Transaction Detail report.
   - `ErrorDetailReport.CMS` – Enter a specific request ID; request IDs appear in the Transaction Detail report.
7. After running the report, you can save it as a CLG file, export it as a PDF or Word file, or print it.

Entering Start and End Times

When running a report, you need to ensure that when you are entering the **Start Date & Time** and **End Date & Time** for the **Reporting Period,** that the time values you enter will include the requests you are searching on. This may mean you will have to increase the end time value (or decrease the start time value) by at least one minute.

For example, if you entered an end time of `14:08` (corresponding to 2:08 pm), then any requests that occurred *one second or more* after 2:08 would not appear in the report. Therefore, to ensure requests that occurred *after* 2:08 are included in the report, you would need to enter an end time of at least `14:09` (2:0**9** pm).

INDEX