Oracle® Insurance IStream

IStream Document Manager Technical Guide

Release 6.3

E15015-01

June 2009

ORACLE

INSURANCE

Copyright

Primary Authors: Andrew Brooke and Ken Weinberg

CONTENTS

Chapter 1 — Overview 11

Document Conventions 12

IStream Document Manager 13

Overview of the Workflow Process 14

Diagram Components 14

IStream Document Manager Components 15

IStream Document Manager Interface 15

Client Components 15

IStream Author 15

IStream Assembler 15

IStream InfoConnector 15

Web Browser 16

Server Components 16

IStream Document Manager Server 16

IStream Assembly Server 16

Additional Information 16

Word Processors 16

Security 17

The System Model 18

IStream Document Manager Server 18

IStream Document Manager Assembly Server 19

Roles 20

IStream Document Manager Documentation 21

Chapter 2 — Getting Started 23

Required Skills 24

Support Skills 24

User Skills 24

System Requirements 25

Chapter 3 — Installing IStream Document Manager 27

Deploying IStream Document Manager 28
Author Workstation 28
Workstation Configuration 28
IStream Document Manager Server 29
Assembly Server 29
InfoSources 30

Planning a New Installation 32
Installation Considerations 32
Configuration of InfoSources 32

Chapter 4 — System Administration 33

Security and Permissions 34
Admin and Permissions 34
Security Model 34
Session Management 34
Workstation Applications 35
Unified Model for IStream Document Manager 35
Security Information Propagation 36

Proxy User IDs 37
IDs Used in Processing 38
Web 38
Author 38
Assembler 38
Notes 38
Security and Permissions with Proxy IDs 38
Errors 39

User Administration 40
Users 40
Groups 40
Listing Users and Groups 40
Creating a New Group 40
Editing a Group 41

Deleting Groups 41
User Profiles. 41
Creating Users 41
Editing and Deleting Users..... 43

The Administration User 44
IStream Document Manager Items 44
Enterprise Workspace Structure 44
Adding IStream Document Manager Items 44
Item Properties. 44
Working with IStream Document Manager Items. 45
Categories and Attributes. 45
Configuring IStream Document Manager Parameters. 46
IStream DM InfoSource Name 46
IStream DM API Log 46
IStream DM Explorer..... 47

InfoSource Administration 48
Defined Variables. 48

Assembler Administration 49
Making Assembly Server Available for Generation 49
Setting Up Generation Mode 50
Running Assembly Server as an Application or Service 51

Optimizing for Performance 52

Chapter 5 — Technical Notes 53

The Log Viewer 54
Opening the Log Viewer. 54
Log Message Types. 55
Using Message Filters 56
Viewing Log Files. 56
Changing the Logging Level 56
Logging Function 57

Document Caching 59
Caching Documents for IStream Assembler 59
About the Cache Folder 60
Cache Folder Maintenance 60

About Cached Files 61
Synchronization of Cached Files 61
Document ID and Version Number 61
Limitations on Document Caching 61

Plug-ins 63
Audit Trail 63
Querying the Audit Trail 64
Purging the Audit Trail 64

Promoting Documents 65
The Model Document Development Process 65
Using Environments 65

Identifying the Software Version 66

Reserved Words 67

Edit and Author Functions 71
Opentext.ini 71

The IStream Document Manager Servlet 72
Servlet Dependency 72
IStream Document Manager Servlet Configuration 73
CalligoServlet.configure 74
InteractiveServlet.configure 74
NonInteractiveServlet.configure 76
IStream Document Manager Servlet Interface Parameters 78
InteractiveServlet 79
NonInteractiveServlet 81
Generating With IStream Publisher 82
CalligoServlet.Generate 82
NextURLSuccess example 84
Invoking CalligoServlet.Generate (No key data) – HTML Example 84

Load Balancing 87

PDF Rendering Properties 88
Automatic TOC Update Registry Setting 88
PDF Minimum Margins 88

Configuring PDF Font Embedding Options 89

Configuring File Options in the Zip & Download and Print Features. 91

Chapter 6 — XML InfoSources 93

XML InfoSources 94
Architectural Overview 94

Structured XML InfoSource 96
Preparing the XML File 96
Structuring the XML File. 96
Structuring an XML File for Batch Mode Processing 96
Structuring an XML File for Interactive Mode Processing 97
Representing Data in the XML File 97
Representing Normal (Scalar) Variable Data 97
Representing Recordset Data 98
Common Problems with Representing Data in an XML File 99
Identical Field Names within the Same Recordset 99
Undefined Values in the XML File 99
Batch Mode Processing – Example XML File 100
Explanation 101
Interactive Mode Processing – Example XML File. 102

XPath InfoSource 103
XPath InfoSource Data Example 103
Preparing the XML File 104
Structuring the XML File. 104
Mapping File Structure 105
Query Name 105
Reference Name 106
Lookup Name 107
Complete Mapping Examples 108
Mapping Example 1 108
Mapping Example 2 109
Mapping Example 3 110
Complete XML Source Examples 111
XML Source Example 1 111
XML Source Example 2 111
About XPath 113

Adding XML InfoSources 115

Coding Model Documents for XML InfoSources 116
Querying Data 116
General Syntax 116
Querying All Data In The XML File 117
LOOKUP and CHECKUP Functions 118
Example Model Document Code 119

Author Wizards and XML 120
XML ISC Query Rule 120
XML Variables 120
Lookup and Checkup functions 121
Allowed Values 121

Chapter 7 — IStream Promoter 123

Opening IStream Promoter 124

Creating a New File List 125

Adding Files to the File List 127

Adding Reference Files to the File List 129

Removing Files from the File List 131

Entering the File Destination Path 132

Promoting or Converting the Files 133
Converting Files 133

Viewing the IStream Promoter Log File 134

Configuring IStream Promoter 135
Setting the Log File Directory 135
Setting the Temporary Directory 135
Purging the Temporary Files 135

Chapter 8 — System Tables 137

The System Tables 138

The FORMATS Table 139

The LANGTBL (Language) Table 143

Index 145

Chapter 1

Overview

Welcome to the IStream Document Manager 6.3 Technical Guide.

This technical guide provides information about deploying IStream Document Manager, system administration and troubleshooting.

This chapter describes:

- *Document Conventions* on page 12
- *IStream Document Manager* on page 13
- *Overview of the Workflow Process* on page 14
- *IStream Document Manager Components* on page 15
- *The System Model* on page 18
- *Roles* on page 20
- *IStream Document Manager Documentation* on page 21

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 Document Manager

IStream Document Manager automates document assembly, management and control. Documents can range from complex, multi-page, multi-format contracts, to statements with extensive customization, and even simple correspondence.

IStream Document Manager helps increase productivity, improve customer service, reduce costs, shrink time to market and improve the appearance and consistency of documents.

IStream Document Manager is a tool for complex document creation on a large scale. It uses document management and workflow techniques, client/server and database practices and the Internet, to help create and manage your business documents.

IStream Document Manager merges information from a database located anywhere a connection can be established (using InfoSources) with model documents. During generation of these complex documents, IStream Document Manager uses document compilers, InfoSources, references, rules, functions and variables to bring various data into the document.

IStream Document Manager consists of a number of products and interfaces, including:

- a Document Management System (DMS): includes client/server, web and administration components
- IStream Author: includes authoring wizards
- IStream InfoConnector
- IStream Assembler (client and server components) for document assembly

Overview of the Workflow Process

The following diagram gives an overview of the IStream Document Manager workflow process and includes the various IStream Document Manager components:

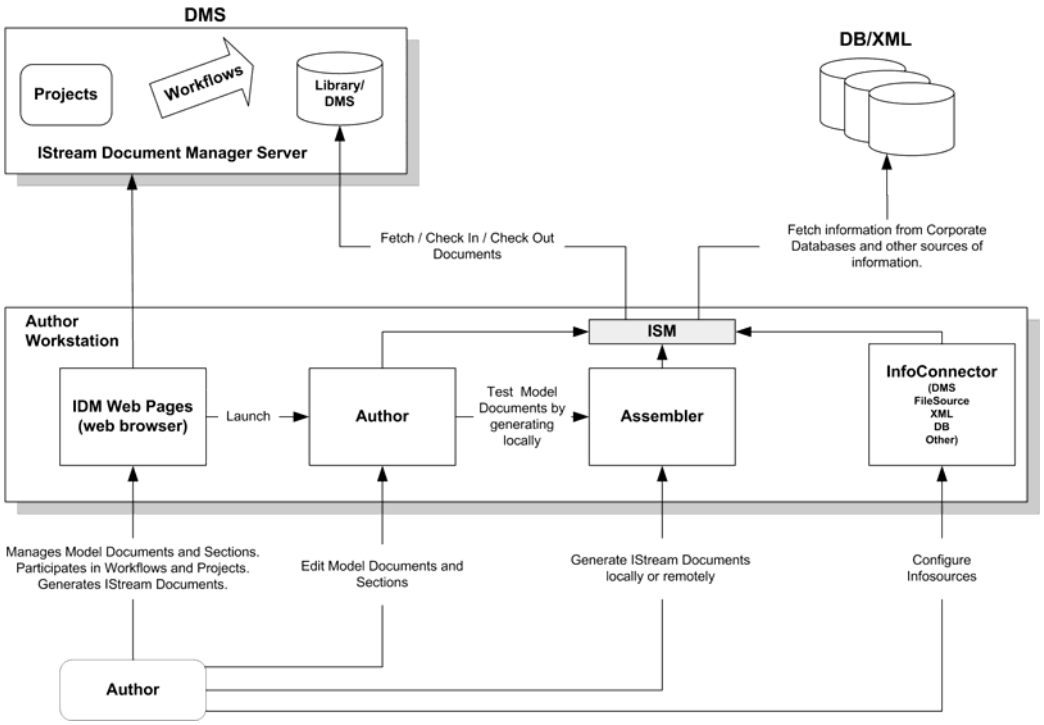


Diagram Components

WorkFlows

Workflows can be created and modified through the browser-enabled workflow editor in the DMS. Its graphical interface lets you see a map of your workflow as you add, remove or edit workflow tasks, milestones, and workflow participants.

For descriptions of the other components, see *IStream Document Manager Components* on page 15

IStream Document Manager Components

IStream Document Manager consists of client components, usually located on a workstation or desktop, and server components, which are usually installed on one or more separate networked computers.

IStream Document Manager has two main interfaces:

- the IStream Document Manager interface which consists of enhanced IStream Document Manager web pages displayed in a web browser: see *IStream Document Manager Interface* on page 15 below
- the **Author**, **Assembler**, and **InfoConnector** interfaces which may also embed Microsoft Word or present a graphical user interface

These interfaces are described in detail in the following sections.

IStream Document Manager Interface

The IStream Document Manager interface provides document and folder manipulation and tracking. It allows you to establish workflows that will better manage your document assets, projects, and other collaborations.

Client Components

IStream Author

This application is used to create, edit and manage model documents and sections. Because it embeds Microsoft Word directly inside of it, editing a section is like editing any other kind of document. It includes tools to simplify the process of authoring, and allows direct interaction with IStream Document Manager or the file system for storing and retrieving model documents and sections. Author can operate independently of the UI or in conjunction with it.

Author includes wizards to help combine rules and functions to construct model documents which will be evaluated during generation to produce complex contracts, policies, and correspondence.

IStream Assembler

Assembler is used to control the generation of model documents to create new IStream Document Manager documents (.clg files). Users will access Assembler functions from the IStream Document Manager if they have the required permissions. Assembler may also be launched from a workstation desktop, or from the Author menu.

IStream InfoConnector

InfoConnector allows authors and system administrators to configure multiple data sources as InfoSources for IStream Document Manager. This allows you to use the business knowledge already in various forms at your site.

You must access InfoConnector separately. For more information, see the *InfoConnector Online Help*.

Web Browser

Users access IStream Document Manager with a web browser by entering the URL for a web page hosted on an IStream Document Manager server. Users must log in before they can do any work. Once they have entered the system, they can navigate the Enterprise Workspace, participate in workflows or projects, and generate IStream documents. The web page they see is in the IStream Document Manager user interface.

Server Components

IStream Document Manager Server

This is the main server for an IStream Document Manager installation. It works with the Web server to provide the IStream Document Manager user interface through the Web browser. It also provides the background services of document management, workflow, project management, administration, and access to document generation through an Assembly Server.

The IStream Document Manager server is on the same computer as the Web server and requires a connection to a database server to handle its data storage requirements.

IStream Assembly Server

This server does the actual work of generating IStream documents. To generate or assemble documents using IStream Document Manager, there must be one or more Assembly Servers available to the IStream Document Manager server through administration.

An Assembly Server can be used by one or more IStream Document Manager servers and each IStream Document Manager server can use one or more Assembly Servers.

Additional Information

Word Processors

IStream Document Manager supports Microsoft Word. It will recognize a section created or edited in Microsoft Word.

Note: Previous versions of IStream Document Manager might not support newer versions of Word.

Security

IStream Document Manager's security features include levels of access for users, model document authors and system administrators. Security procedures ensure that the documents and databases are protected from internal, external or Internet interference.

The System Model

The physical layout of an IStream Document Manager configuration is a set of servers and clients that include:

- a web server that represents the IStream Document Manager server host location
- a DMS server located on the host computer for storing and controlling documents and providing the main user interface through web pages
- one or more Assembly Servers located on different remote computers, the host computer, or both
- a RDBMS server
- workstations or clients distributed throughout your enterprise

Note: All users must be running a web browser and may have one or more desktop applications installed.

IStream Document Manager Server

The IStream Document Manager server involves the following areas:

- user and group administration and security enforcement
- registration and access to administration of secondary servers
- the IStream Document Manager home page, which includes the Enterprise Workspace, individual account folders and other areas

IStream Document Manager provides all the document repository services as well as the user interfaces to these services. This includes:

- document storage and retrieval
- search capability
- container structures, such as folders, compound documents, and environments
- navigation through the UI
- document profile management (categories, attributes)
- permissions (user and group IDs come from the IStream Document Manager server)
- DMS administration
- assembly execution
- workflow map creation and editing
- workflow initiation method and execution engine
- monitoring and administration

IStream Document Manager Assembly Server

The IStream Document Manager Assembly Server provides the following services:

- generation (assembly) of new documents or regeneration of existing documents
- logging of process messages to generation log

Roles

The following table outlines the various roles for users of IStream Document Manager:

Role	IStream Document Manager Components	Business Process
End User	• Assembler • DMS • workflows - personal workspace • collaborative projects	• Document issuance process • Document search and retrieval
Author	• End user components when testing • Author • Assembler • DMS (document management and DMS admin tools) • InfoConnector • Authoring wizards	• model document creation, development and change processes
Administrator and Technical Support	• End user components • Assembly Server (admin) • DMS (admin) • Workflow (admin)	• system management • user and group management

IStream Document Manager Documentation

IStream Document Manager includes the following documents and online help files:

General Documentation

- The *IStream Document Manager 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 Document Manager ReadMe* file describes the contents of the installation package.
- The *IStream Document Manager Glossary* contains definitions of commonly used IStream terms.

User Guides and Online Help

- The *IStream Assembler Online Help* describes how to generate documents from sections and model documents.
- The *IStream Assembler Add-in for Microsoft Word Online Help* describes how to perform certain IStream Assembler functions directly within Microsoft Word.
- The *IStream Author User Guide* describes how to create, edit, and test sections or model documents.
- The *IStream Author Add-in for Microsoft Word Online Help* describes how to perform certain IStream Author functions directly within Microsoft Word.
- The *IStream Customizer User Guide* describes how to modify the content of generated documents so that the changes will be applied during subsequent document generations.
- The *IStream Document Manager DMS Guide for IStream and Model Documents* describes how to edit and work with IStream sections and model documents using the DMS user interface.
- The *DMS Plug-in for Author Online Help* describes how to access DMS functionality in IStream Author.
- The *IStream InfoConnector Online Help* describes how to set up and configure IStream InfoSources.

Advanced Guides

- The *IStream Document Manager Guide for New Installations* explains how to complete a new installation of IStream Document Manager. It includes system requirements and detailed installation and configuration information.

- The *IStream Document Manager Upgrade Installation Guide* explains how to upgrade an *existing* installation of IStream Document Manager. It includes system requirements and detailed installation and configuration information.
- The *IStream Document Manager Message Reference Guide* contains lists of error, log and SDK messages from the various IStream components. It is for technical users who need additional information about the various IStream messages they receive.
- The *IStream Document Manager Technical Guide* is for system administrators and technical support staff who configure IStream Document Manager and set up links between it and their company's database. This guide is also for those who set up and maintain security groups and operators in the system, and who install, optimize, maintain and troubleshoot IStream Document Manager. It also describes installing and configuring IStream XML InfoSources for interactive and batch generation.
- The *IStream Assembler* and *Extensibility Toolkit Guides* contain detailed descriptions of the components in the IStream Toolkit, or SDK. They are for technical users who need to integrate IStream components with their own or other third party applications.

  In addition to the toolkit guides, the *IStream Toolkit SDK Samples Guide* gives an overview of the toolkit samples. The samples are working examples that can help you develop custom applications.

  For more information about the toolkit, see the *IStream Document Manager Toolkit ReadMe.*

Chapter 2

Getting Started

This chapter describes:

- *Required Skills* on page 24
- *System Requirements* on page 25

Required Skills

IStream Document Manager users and those who will be supporting IStream Document Manager should have experience in client/server technology, databases and web services.

Support Skills

Your IStream Document Manager technical team should have the following skills to help you configure and optimize your system:

- **operating system**: Microsoft Windows
- **programming languages**: visual basic, visual C++, COM.
- **database skills**: database management, administration and design; knowledge of tables and queries
- **communications**: TCP/IP and networking
- **HTTP Server**: Microsoft Internet Information Server
- **word processor**: Microsoft Word

User Skills

Your users should have the following skills:

- advanced Microsoft Word skills for authors
- familiarity with web browsers and hypertext
- understanding of simple database structures

System Requirements

Note that all server side software may reside on a single host or you can install the IStream Document Manager server and web server on one system with the RDBMS on another system. The Assembly Server can be on a separate computer or the same computer.

The second option will provide greater capability, but you must also make sure that the necessary RDBMS client software resides on the IStream Document Manager server. The web servers must always reside on the same host as the IStream Document Manager server.

For the specific system requirements, see the *Check the System Requirements* on page 20 of the IStream Document Manager *Installation Guide.*

See the *Release Notes* for the specific software and versions required.

Other requirements include:

Databases (Data Sources)

IStream Document Manager supports industry standard ODBC databases, including:

- Microsoft Access
- Microsoft SQL Server
- Oracle

For support of a specific database not listed above, or a specific RDBMS version, contact customer support.

Other InfoSource Types

- file system
- IStreamDM
- IStreamXML
- DynaFS – this is for internal use only; do not change or remove

Other InfoSources (non-ODBC) can be supported by creating a proprietary InfoSource type. See the *IStream Document Manager Toolkit* (SDK) documentation for details.

Chapter 3

Installing IStream Document Manager

Complete installation instructions are provided in the *IStream Document Manager Installation Guide*. However, this section contains additional information which should help you better prepare the installation and understand how the IStream products integrate with each other and your existing network. This information may also help you solve problems or train other system administrators to work with IStream Document Manager.

This chapter describes:

- *Deploying IStream Document Manager* on page 28
- *Planning a New Installation* on page 32

Deploying IStream Document Manager

The deployment of IStream Document Manager is not a one-step process. There are different bundles of IStream Document Manager components which may be installed on one or more servers and workstations. Certain components must be installed before other components. For more information, see the *IStream Document Manager Installation Guide*, including *IStream Document Manager Features* on page 10. This section lists the various *IStream Document Manager* components that are installed.

Author Workstation

The Author workstation is used to create and edit model documents and generate documents. All IStream Document Manager functions should be available to the author from the Author workstation. The installation program will install the following components on the Author workstation:

- AppLauncher
- Assembler
- Assembler client
- Author
- Authoring Assistance
- IStream Document Manager Client
- IStream Document Manager API
- IStream Document Manager InfoSource
- IStream Document Manager Plug-ins
- FileSystem InfoSource
- IStreamXML InfoSource
- InfoConnector
- ODBC Database InfoSource
- Repository
- UserDB InfoSource
- WPSupport

Workstation Configuration

The following components are configured:

- a default file system repository
- the word processor used for logs: the path to a word processor (such as Note Pad or WordPad) is set during the installation
- Assembler options: local

IStream Document Manager Server

The IStream Document Manager server contains the components required for the DMS. The following components are installed:

- IStream Document Manager server
- IStream Document Manager components (the HTML files and OSpaces)
- a Start up menu item to the Administration pages

Assembly Server

The Assembly Server provides remote document generation capabilities. The following components are installed on the Assembly Server:

- Assembler
- Assembly Server
- IStream Document Manager API
- IStream Document Manager InfoSource
- FileSystem InfoSource
- InfoConnector
- IStreamXML InfoSource
- ODBC InfoSource
- Repository
- UserDB InfoSource
- WPSupport

InfoSources

Assembly Server requires InfoSources for certain generation processes. The IStream Document Manager installation process creates and sets up the following required InfoSources:

Name	Type	Description	Configuration Details
system (IStream InfoSource)	ODBC	IStream Document Manager system tables, System InfoSource	**Data Source**: System **Application**: None See *Configuring System InfoSources on page 30.*
IStreamDM (IStream InfoSource)	IStreamDM	the root of IStreamDM; a pre-installed InfoSource	The **Server Name**, **Port** and **Location** should be set according to the installation of DMS server. Do not change the **Location** because this will cause web generation from the DMS to fail. **Note:** This InfoSource is available only if you have an IStream DMS license. Otherwise, the IStream File System (listed next) will be used.
IStreamFS	File System	the default, pre-installed IStream Document Manager file system InfoSource	**Location:** `[IStream Document Manager install folder]\IStreamDocuments`
DynaFS	(Internal use only)	used by IStream Document Manager for internal purposes	

Configuring System InfoSources

For the system infosource, the data source should be installed and configured to use 32-bit ODBC drivers (Microsoft), using the name **system**. The system tables (`formats`, `langtbl` and `office`) delivered in Access 2000 format should be copied into your RDBMS. By default, they will be installed using the Access ODBC driver to point to the system tables. If you move them to your RDBMS, the appropriate ODBC driver should then be used to create an ODBC datasource (DSN) called **system**, which points to the location of those tables. Note that system is the **default** name, and you can change it.

Installing InfoSources

These InfoSources should be installed on the same host computer where the Assembly Server will be installed, and also on the Author workstation. This means that the `LOCAL.IDB` file containing the description of these InfoSources will be installed on the host computer.

At the same time, the directory where it will be installed should be made accessible through the network. That is, it should be possible to reference this file through a UNC, although this is not mandatory.

Note that these InfoSources are configured automatically during the deployment of IStream Document Manager.

Planning a New Installation

When you are preparing to set up a new IStream Document Manager installation, design the structure of your users and groups first, making base groups and creating larger groups out of smaller groups.

Before installing IStream Document Manager, you need to determine:

- who should be able to create and edit user profiles, other than the administrator
- the audit options
- how you intend to set proxy userIDs: see *Proxy User IDs* on page 37
- where to install each IStream Document Manager component: *IStream Document Manager Features* on page 10 of the *IStream Document Manager Installation Guide*

Installation Considerations

You can install some components on one or many computers but note that:

- if the RDBMS is on a separate computer, ensure that you can connect to it
- if the database server is on a separate computer, you need the database client software for that particular server installed on the IStream Document Manager Server and any computer doing document assembly
- if the IStream Document Manager server is on a separate computer, ensure that you have TCP/IP connectivity to it

Also note that if you have more than one registered version of Word, or if you have Word in the registry more than once, you may encounter problems.

Configuration of InfoSources

After installing the Author workstation, you must configure any additional InfoSources that you require, other than the ones installed during deployment. To do this, launch the InfoConnector, which is used to create your InfoSources.

Create any other InfoSource that you want to use during local generation.

Chapter 4

System Administration

This chapter describes:

- *Security and Permissions* on page 34
- *Proxy User IDs* on page 37
- *User Administration* on page 40
- *The Administration User* on page 44
- *InfoSource Administration* on page 48
- *Assembler Administration* on page 49
- *Optimizing for Performance* on page 52

Security and Permissions

Admin and Permissions

You can use the IStream Document Manager **Administration** and **Permissions** interface to manage users, groups and individual permissions. The IStream Document Manager administration extension for defining Assembly Servers allows an administrator to specify:

- which Assembly Servers a particular user has the right to use
- what credentials (the proxy userID and password) that user is allowed to use when assembling on a particular server

The administrator can assign special different credentials to use during assembly. One proxy user ID can be used as a user ID for a set of users. This proxy is set up on all information sources that are accessed during assembly, such as databases and repositories. This prevents the need to set up every user on every information source independently: see *Proxy User IDs* on page 37

Security Model

Session Management

A client typically accesses a server in the following process, which occurs in the background:

1. The client establishes a connection with the server, for example, a TCP/IP socket.
2. A client can only issue commands to the server in the context of a server session. If the client does not already have a session, it attempts to acquire one from the server. If the server has a security setup, the client logs in using the appropriate credentials (userID/password or proxy). The server attempts to authenticate the credentials. If they pass, then a session is created for the client.
3. The client logs out, releasing the session.
4. The client disconnects from the server.

Connections and Sessions Durations

There are two scenarios regarding the durations of connections and sessions:

- The duration of the session can equal the duration of the connection. A new session is established after a connection is made, and is removed when the connection is lost. Web browsers or servers and IStream Document Manager use this approach.
- The session can exist beyond the connection and can therefore be used in multiple connections. Assembler uses this approach.

Because connections and sessions take up resources, in interactive contexts, they are typically assigned as needed, and then released. That is, whenever the user

wants to perform an operation, they log into the server, obtain a session, perform the operation and then log out releasing the session. In batch operations, it is more efficient to hold a session for the entire batch.

Workstation Applications

Workstation applications and local generation do not require a login and can be launched by anyone who has access to their own workstation. Security is applied to these components only indirectly. This is because to use these components requires access to documents (in a DMS) or information sources (for generation) which may be controlled by security.

Unified Model for IStream Document Manager

Using this basic security model for client/server and application components, users will see that:

- workstation applications do not require security, however, when a document is opened from a secure information source, the user must log in
- after logging in, the connection to the information source remains while the application is active
- when a server operation is initiated through a client, a hostname is required to establish a connection to a server and a login will be required to gain access; the user must also provide the database connection (database name) to proceed.
- access to IStream Document Manager using the browser requires a URL and then a login using a web page
- access to a generation server (using an application or web page) requires that the application (or web page) pass the hostname and password, and the credentials required by the generator to access information sources during generation.

This model provides the basic security on information and services, but does not provide a unified view of security for IStream Document Manager as a whole. In a unified IStream Document Manager, the user will see that:

- they log in once to IStream Document Manager and have all their application and server security needs captured by this single login
- they cannot use the DMS functions of IStream Document Manager until they log in
- an administrator can create user IDs and set permissions in a single place, and the permissions extend to all security controllable aspects of IStream Document Manager

Security Information Propagation

The mechanism used in IStream Document Manager to create a unified security model is based on the propagation of security information. In this scheme, once a user has successfully logged into the main security point, the security information required by different servers is passed between user interface points (web pages and applications). In this process:

- the user does not have to enter their userID and password each time they reach a security point, because these are passed between applications and servers and used on the user's behalf.
- the main security point can control the security information used by the user at the other security points

When a user initially logs into IStream Document Manager using a web browser, their credentials are authenticated and then stored in their browser. This login remains in effect until the user logs out, or until the browser, is closed which forces a logout. While users are logged in, they can click links and perform operations. All of the subsequent authentication during a user session is carried out in the background.

When Assembler is initiated, the following security information is obtained from the Serverinfo and Userinfo:

- the generator hostname (identifying the server to use) and port
- the credentials (userID/password or proxy) that will be used to access information sources during generation
- if a user opens a model document in the DMS through Author, the credentials found in the user information of the model are maintained while in Author

Proxy User IDs

There are two types of user IDs in IStream Document Manager: a particular user's ID and proxy user IDs. When a user logs in to IStream Document Manager, the login ID is recognized as unique to that user: this is the userID.

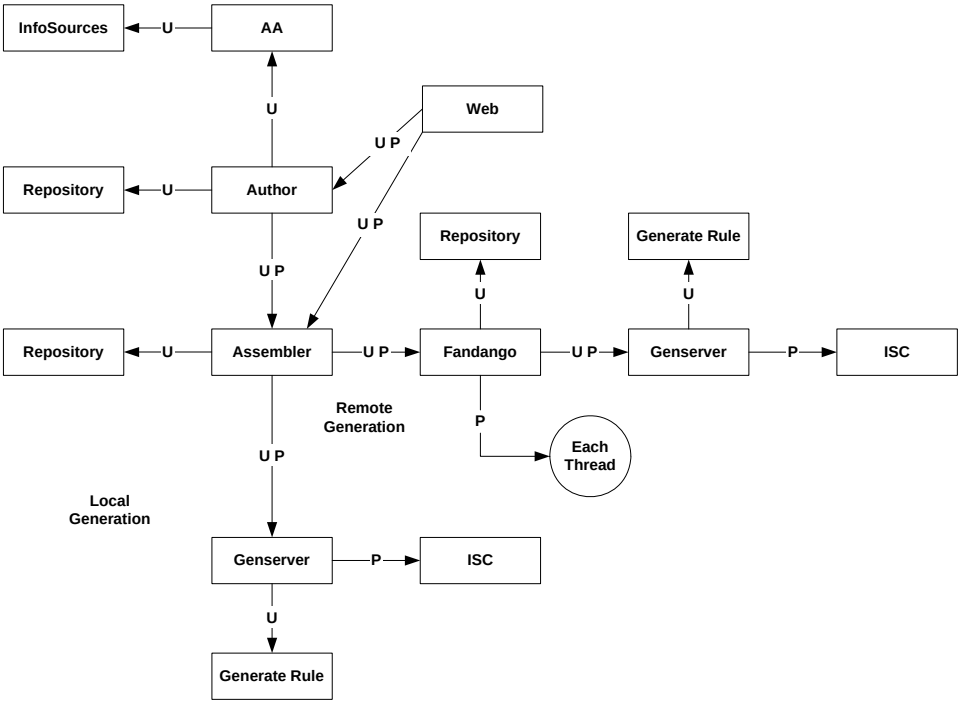
Proxy IDs are special IDs which are assigned to a group of users, or to one or more users to perform a specific set of tasks. IStream Document Manager recognizes proxy IDs so that you do not need more than one ID for each user.

Proxy IDs improve system performance and make it easier for users to do certain tasks by reducing the number of times a user must log in to a database or the Assembly Server. If proxy IDs were not used, then all IStream Document Manager users would need to be set up as users of all the databases required, and for each Assembly Server.

Proxy IDs are associated with the Assembly Server and are set up as its permissions by way of the **Specific Properties** menu of a selected Assembly Server in the DMS.

With proxy IDs, an administrator may assign a number of users to a group and then assign a proxy ID to that group. IStream Document Manager then uses this proxy ID when an information source, such as a database, is accessed. The administrator must register this proxy ID as a valid user at each information source, but one ID can now serve a number of users. Security and permissions are still in place when using proxy IDs: see *Security and Permissions with Proxy IDs* on page 38.

The following diagram shows how the user IDs and proxy IDs move through IStream Document Manager and where they are is used:



IDs Used in Processing

The following sections describe each component of IStream Document Manager uses user IDs.

Web

Each time Author or Assembler are launched, the relevant user ID and proxy user ID and passwords are extracted from the DMS and incorporated into the .ALF scripts that are used to launch Author and Assembler.

Author

Author needs to accept both a user ID and password, and a proxy user ID and password, if it is launched from the web client. Author uses the user ID and password to access the repository, if the repository requires these credentials.

Author passes only the user ID to AA (Authoring Assistance). Author passes both the user ID and password and the proxy user ID and password to Assembler only if Author was launched from the web client.

Assembler

Unless it is started from the desktop, Assembler needs to accept both a user ID and password and a proxy user ID and password. Assembler uses the user ID and password to access the repository. Assembler uses the proxy user ID and password to access InfoSources. Assembler passes the user ID and password into each call to the `Generate` rule.

Notes

In general, for every component which may accept both a user ID and a proxy user ID, the user ID and password must be supplied. If the proxy user ID and password are not supplied, then these should use the same values as the user ID and password.

The system remembers the user's real user ID and password as well as the proxy ID, and sends both through because different components need different IDs to approve processing.

Security and Permissions with Proxy IDs

Proxies are assigned for specific purposes, and the permissions are restricted to that assignment. User IDs are used to access documents, therefore security and permissions for documents will be in accordance with what has been set for a particular User ID or group (if that UserID is a member of a group).

Proxy IDs are used to access data in a database, therefore a user with more restrictive privileges than those associated with a proxy ID will *not* have increased privileges simply by using a proxy ID.

Errors

If you notice errors such as being unable to locate documents, the proxy IDs may not have been set up correctly.

User Administration

The Admin (Administrator) user sets up the permissions and associations for all users and groups on the system. This includes giving other users permission to create and maintain groups and to perform administration functions.

Administration functions are accessed in the DMS from the **Users & Groups** page from the **Enterprise** menu. In the **Users & Groups** page is open, select an item from the **Search** drop-down list box, enter a value in the adjacent field, and then click **Find**.

Users and groups are used in defining access permissions and workflow steps for routing tasks. For additional information on access permissions, see *Proxy User IDs* on page 37.

Users

Users establish an identity for everyone who logs in and connects to an IStream Document Manager server. Everyone who regularly uses the IStream Document Manager system should their own user login.

Groups

A group is a set of IStream Document Manager users and/or other groups. Each user can be a member of more than one group. One of the groups is considered the user's *base group*: the group of people with which the user works most often.

Listing Users and Groups

All IStream Document Manager users can see the list of users and groups. Whether you can view or edit a group depends on your privileges and whether you created the user or group.

Creating a New Group

1. From the **Enterprise** menu, select **User & Groups**.

   The **Users & Groups** page opens.

2. Select **Group** from the **Add Item** drop-down list in the top right of the page.

   The **Add New Group** page opens.

3. Enter a unique group name for the new group in the **Group Name** field and click **Submit**.

   The **Edit Group** page opens, displaying the new group's name in the header.

4. Use the **Search** fields at the top of the form to list the users or groups you want to include in the group.

5. Click **Find**.

   A list of users or groups or both appears.

6. Select the **Add to Group** check boxes adjacent to the names of those users or groups you want to add.
7. Click **Submit**.
8. Click **Done** in the left pane.

Editing a Group

1. From the **Enterprise** menu, select **User & Groups**.

   The **Users & Groups** page opens.
2. From the **Search** drop-down menu, choose **Group Name** and enter the name of the group in the adjacent field.
3. Click **Find**.

   A list of groups appears.
4. Click **Edit** for the group you want to edit.
5. Click the **Find and Add** link to add members and repeat steps 4 to 7 of *Creating a New Group* on page 40.
6. To change the group name, click **Edit Group Name** and enter a new name for the group.
7. Click **Done**.

Deleting Groups

1. From the **Enterprise** menu, select **User & Groups**.

   The **Users & Groups** page opens.
2. From the **Search** drop-down menu, choose **Group Name** and enter the name of the group in the adjacent field.
3. Click **Find**.

   A list of groups appears.
4. Click **Edit** for the group you want to delete.
5. Click **Delete Group**.
6. Click **OK** to confirm.

User Profiles

Each user has a profile which includes the user's privileges and other basic information. You can only create user profiles if you have the necessary permissions. Before creating profiles, you should create base groups for those users, or verify that the groups you want to use exist.

Creating Users

1. From the **Enterprise** menu, select **User & Groups**.

The **Users & Groups** page opens

2. Select **User** from the **Add Item** drop-down list in the top right of the page.
3. Enter the user information:
   - **Login name** – this is only case sensitive by default for an Oracle database)
   - **Department** – the group they belong to
   - **Password** – case-sensitive
   - **Verify password** – case-sensitive
   - **First name** – descriptive
   - **Middle initial**
   - **Last name**
   - **Title**
   - **E-mail**
   - **Phone**
   - **Fax**
   - **Office Location**
   - **Time Zone**

   Privileges:

   - **Log-in enabled**
   - **Public Access Enabled**
   - **Can create/modify users**
   - **Can create/modify groups**
   - **User administrator access**
   - **System administration rights**
4. When you are done, click **Submit** to save your changes.

Editing and Deleting Users

If you have User Administrator access, you can edit or delete *any* user or group. Otherwise, you can only edit and delete the users and groups that *you* created, if you have the **Create/Modify** privilege.

1. From the **Enterprise** menu, select **User & Groups**.

   The **Users & Groups** page opens.

2. Search for the **User** or click **Find** to see all users.

3. Click the **Edit** link for the user you want to edit.

4. Change the information as necessary, then click the **Update User** button.

   To delete the user, click **Delete User**.

The Administration User

The Admin user login has special privileges and responsibilities within IStream Document Manager. When IStream Document Manager is first installed, Admin is the only user.

The interface for much of the functionality available only to the Admin user is separate from that of the standard end-user.

The following URL is used to access Admin functions on the DMS server:

server_name/service_name/`livelink.exe?func=Admin.index`

IStream Document Manager Items

As the Admin user, you can set privileges for many of the IStream Document Manager items in the Enterprise Workspace. In particular, you can restrict access to folders, environments, documents, and projects, as well as certain functions which act on these items, such as generation capability and **Add Version**.

Enterprise Workspace Structure

The Admin user defines the basic hierarchical structure of the Enterprise Workspace, and is at the top of this structure. Access permissions, as well as categories and attributes, are copied from folders to their contents (or inherited).

Note: The creation and management of folders is critical to business processing and should be carefully planned.

Adding IStream Document Manager Items

1. Navigate through the **Enterprise Workspace** structure to the folder or environment where you want to add the item.
2. From the **Add Item** drop-down menu, choose the item type you want to add.

Note: If you do not see the **Add Item** menu, you do not have permissions to add items to the **Enterprise Workspace**.

3. In the **Add** page that appears, enter a name for the item and any other relevant information.
4. Click **Add** to complete the operation.

Note: Names must be unique within a folder, otherwise a new version of an existing document will be created.

Item Properties

Every object in IStream Document Manager has a set of properties that can be viewed and sometimes edited from its **Properties** page.

Working with IStream Document Manager Items

Users work with IStream Document Manager items in different ways depending on their permission levels. All of these options can be accessed through the **Functions** menu on the **General Info** page for each item:

Categories and Attributes

Categories and attributes of items within IStream Document Manager allow you to control how the system handles and locates documents and other items. In particular, categories and attributes provide for complex searches across your DMS and prove metadata or additional information about each item.

A **category** is a classification, type, or grouping of category or documents and other items.

An **attribute** is a field or data element which describes a document or other item, serving as metadata to use in searches.

You can create your own custom categories and attributes by using the **Admin** page in IStream Document Manager. There are basic attributes assigned to each item, such as who created it, the creation date, and its size.

As the Admin user, you can make attributes for each item be required or optional as they are created.

Categories include:

- the category name
- whether to include attribute
- whether the attribute is required: select both the **Included** and **Required** checkboxes for a required attribute

Attributes include:

- keywords
- name
- field name
- type/data type
- text length; length
- display name
- default value

Configuring IStream Document Manager Parameters

This section describes the parameters that can be changed in IStream Document Manager.

Method: Configure IStream Document Manager parameters

1. Open the IStream Document Manager administration page: *server_name/service_name/*`livelink.exe?func= Admin.index`
2. On the IStream Document Manager **Administration** page, go to the **IStream Document Manager Administration** section
3. Click the **Configure IStream DM Parameters** link
4. On the **Configure IStream DM Parameters** page, change any of the following parameters:
   - *IStream DM InfoSource Name* on page 46
   - *IStream DM API Log* on page 46
   - *IStream DM Explorer* on page 47

The following sections describe these parameters.

IStream DM InfoSource Name

This field defines what InfoSource name the DMS should use when it constructs a document UISR when a user opens Author, Assembler, or Customizer through Internet Explorer, and has a different InfoSource name for their Livelink DMS.

Example: A user has an additional IStream Document Manager InfoSource named `IStream2` in their InfoConnector. If this name is still `IStreamDM` in the DMS, and the user opens a document through a menu in Internet Explorer, the UISR they will get will be `IStreamDM:path` instead of `IStream2:path` and the operation will fail. To prevent this, `IStream2` must be entered in the IStream Document Manager InfoSource **Name** field.

Important: The value of this field should always be `IStreamDM` and should only be changed if instructed to do so by Customer Support.

IStream DM API Log

When selected, this checkbox enables extra DMS logging for LAPI function handles on the DMS side. It also enables extra logging for the entire DMS. You can select this checkbox to do debugging when you want to view output values sent to the LAPI call made from the client or view other diagnostic data.

IStream DM Explorer

When selected, this check box enables Livelink Explorer to retrieve IStream documents (.CLG), master sections (.CMS), and sections (.CDS) correctly instead of Word documents (.DOC) stripped of any IStream content.

If this check box is cleared, Livelink Explorer will only retrieve the Word document (.DOC) version with no IStream content.

InfoSource Administration

IStream Document Manager uses information sources (called InfoSources) primarily during model document creation and generation. Connections to various information sources are made, data is retrieved and inserted into the generated document, and then the connections are closed.

Model documents provide the instructions about which information source will be contacted, what information will be obtained, and how that will appear in the final generated document. This information is provided through the use of InfoSources within the model documents. When authors create model documents, they use UISRs, which are made up of InfoSources and references to various data items such as other sections and database fields.

Through IStream Author, IStream Assembler or IStream Customizer, a user may access data from any of these information sources. File system and IStream Document Manager InfoSources represent repositories or document libraries where model documents, sections, and assembled documents are stored.

You configure and manage InfoSources using the InfoConnector application.

Note: For complete information about InfoConnector, InfoSources, and the supported InfoSource types, refer to the *InfoConnector Online Help*.

Defined Variables

Defined variables, or a local variables pool, can be considered as temporary storage of a set of variable triplets (name, type and value), created and retrieved in the process of document generation. These are not configured as InfoSources through InfoConnector, but are provided in the list of available InfoSources accessible through the Reference, Rule and Function wizards.

Assembler Administration

This section describes administering the Assembly Server.

Making Assembly Server Available for Generation

To generate from the Web, at least one Assembly Server must be installed and this computer must be made available to the IStream Document Manager server.

To make an Assembly Server available:

1. In the Enterprise Workspace, go to any environment folder (or create one).
2. From the **Add Item** drop-down menu, choose **Assembly Server**.
3. Type a name for the Assembly Server and click **Add**.
4. Click the **Properties** > **Specific** link for the Assembly Server.
5. Enter the following information:
   - **host address** (the name of your server)
   - **port** 4300 (default)
   - **enabled** (choose this box if you want this server to be enabled for generation)
   - **proxy user ID** and **password** (used for accessing databases or files during generation)

To limit access to this server to specific groups or users (by default, all groups and users have access to a newly installed Assembly Server), you need to complete these steps:

1. Select the Assembly Server to change the permissions of.
2. From the **Function** menu, choose **Permissions**.
3. Edit the public access permissions. If you want everyone to see this Assembly Server, the **See** and **See Contents** setting must be on. To prevent public access, disable all permissions.

Note: To add or generate IStream documents, users must have at least the **See** and **See Contents** permissions to the active Assembly Server.

4. To give access to specific groups or users, choose **Grant Access** and from the **search** menu, select the users or groups you want to give permissions to.

Setting Up Generation Mode

IStream Assembler has a configuration parameter called the Generation Mode, which controls how the information about model document tags and custom wordings are processed during document generation. IStream Assembler uses this parameter in cases when the client application does not pass this parameter to IStream Assembler together with the request for document generation.

Customizer passes the value of the Generation Mode to the IStream Assembler when the user invokes the **Generate** function from it. Any custom application using the IStream Document Manager SDK can also pass the Generation Mode to the IStream Assembler.

The following table explains the Generation Mode:

Gen. Mode	Description	Generated Documents	Regenerated Documents
0	Standard mode	Cannot be customized.	Do not contain the original custom wordings, have no tags and cannot be customized.
1	Standard mode	*(Do not use this mode for generating. Use it only for regenerating.)*	Contain the original custom wordings and cannot be customized.
2	Customizer mode	Contain tags and can be customized.	Do not contain the original custom wordings, contain tags and can be customized.
3	Customizer mode	Contain tags and can be customized.	Contain the original custom wordings, contain tags and can be customized.
15	Customizer mode	Will contain tags and the intermediate data from the last generation to enable the QuickGen feature in Customizer.	Contain the original custom wordings, tags and the intermediate data from the last generation to enable the QuickGen feature in IStream Customizer.

In this table:

- **Gen. Mode** is the actual generation mode numeric value.
- **Description** is a brief description of the mode.
- **Generated Documents...** indicates whether documents generated using this mode can be customized using IStream Customizer, and also what these generated documents will contain.
- **Regenerated Documents...** indicates whether documents regenerated using this mode can be customized using IStream Customizer, what these generated documents will contain, and whether they have QuickGen enabled.

After IStream Assembler is installed, the Generation Mode is set by default to 15, making your documents “QuickGen ready”. However, you can change this setting by completing the following procedure.

Method: Change the Generation Mode

Warning: Only advanced users and system administrators should edit the Windows registry. We recommend you back up the registry before proceeding.

1. Start the Registry Editor on the computer where IStream Assembler was installed.
2. If IStream Document Manager was installed in the default folder, navigate to the following path:
   `HKEY_LOCAL_MACHINE\SOFTWARE\Oracle\IStream_Components\DG\Parameters`
3. Select the **Parameters** node in the tree view, in the left pane of the Registry Editor.
4. In the right pane of the Registry Editor, right mouse-click the **GenerationMode** item, and select the **Modify** command from the pop-up context menu.
5. In the **Edit DWORD Value** dialog box, change the value in the **Value** data entry field to the value you want:
   - 15 – generated documents will be QuickGen ready (default setting)
   - 3 – generated documents will *not* be QuickGen ready
6. In the **Base** section, select **Decimal**.
7. Click **OK** to close the Registry Editor.

Running Assembly Server as an Application or Service

The Assembly Server must start as a Win32 service rather than as a console application. This is the best way to allow a process to have a custom timeout. The Generation Server needs time to release all cached resources, so providing sufficient shutdown time for the Assembly Server is important to release all allocated resources.

Note: A Win32 process cannot be terminated from the Task Manager window.

Operating as a Win32 service, the Assembly Server has some “share” access issues. By default, the Assembly Server is running under the system account, which does not have access rights to any remote “shares”.

Before starting the Assembly Server, check the Log On As: section when you click the Startup... button in the Services window.

For Remote Generation, log on as the same user that you have assigned above.

In a production environment, all Win32 services should be controlled (select **Start** > **Stop** > **Shutdown**, and so on) by the system. No user is required to log on to run the service.

Optimizing for Performance

We have identified some ways to overcome possible performance problems when generating model documents. Here are some suggestions to reduce generation times:

1. Page file size configuration

Page file size on the server causes lengthier remote generation times than necessary. If the page file size reaches its configured maximum limit, then Assembler (iaegensm.exe) produces many more page faults, slowing down the generation process.

We recommend the following system configuration requirement for Assembly Server and IStream Assembler operation:

Initial Size: 75 MBs (Client Workstation)

128 MB (Server)

Maximum Size:128 MB (Client Workstation)

192 MB (Server)

2. Use Caching *(See Document Caching* on page 59).

Chapter 5

Technical Notes

This chapter describes:

- *The Log Viewer* on page 54
- *Document Caching* on page 59
- *Plug-ins* on page 63
- *Promoting Documents* on page 65
- *Identifying the Software Version* on page 66
- *Reserved Words* on page 67
- *Edit and Author Functions* on page 71
- *The IStream Document Manager Servlet* on page 72
- *Load Balancing* on page 87
- *PDF Rendering Properties* on page 88
- *Configuring PDF Font Embedding Options* on page 89
- *Configuring File Options in the Zip & Download and Print Features* on page 91

The Log Viewer

You can use the Log Viewer to view all informational messages, error messages and warnings that are generated from all IStream applications, in a single viewing area. All messages are stored in a series of log files and folders that are named based on the log date and time.

Note that the Log Viewer has its own online help with complete information and instructions. This section of the Technical Guide contains additional information and describes:

- *Opening the Log Viewer* on page 54
- *Log Message Types* on page 55
- *Using Message Filters* on page 56
- *Viewing Log Files* on page 56
- *Changing the Logging Level* on page 56
- *Logging Function* on page 57

Opening the Log Viewer

Use the following procedure to open the Log Viewer and view your current log messages.

Method: Open the Log Viewer

1. Click **Start** > **Programs** > **IStream** > **Logs** > **IStream Log**.

   The Log Viewer opens.

2. When you first use the viewer, the default directory is `C:\Temp`, and not the log directory. Therefore, you will need to browse to the correct location for where the logs are saved. To do this:

   a. Click **File**, then **Open Log**, or click the **Open Log** button.

   b. Navigate to the following folder and open the log file in it:

   `IStream Document Manager_install_folder``\Logs\`

3. Each message appears in the right pane with the following information:

   - the severity **Level** of the message; there are four levels of messages - for details, see *Log Message Types* on page 55
   - the date and **time** that the message was logged
   - **Module** – the program executable (.EXE) file that the message was generated from
   - **Case** – the specific submodule that the message was generated from; this can include various .DLLs

- **Text** – The actual text of the error message. You can double-click the text to read additional information about the message.

Tip: To open the current log file, click **File**, then **Open Recent Log** or click the **Open Recent Log** button.

Log Message Types

There are four levels of messages. From least to most critical, these are:

- **Informational** messages – These appear during the normal operation of IStream Document Manager. They do not indicate any operational problems. By default, these do not appear in the log viewer.
- **Warnings** – These indicate a problem may have occurred.
- **Errors** – These indicate a problem has occurred, but IStream Document Manager may be able to continue running normally.
- **Critical** – These indicate a serious problem has occurred that may cause IStream Document Manager to stop working.

Note: In the log viewer, **Critical** will appear as **Alarms**.

Using Message Filters

You can use filters to display only certain types of messages.

Method: Apply a filter

1. Drag and drop a **Module**, **Case** or **Custom** filter from the left pane onto the right pane.
2. The filter is applied.

Method: Remove a filter

1. Right-click the right pane and select **Release Filter**.
2. The filter is removed.

Viewing Log Files

All IStream Document Manager messages are stored in log files. If this maximum size is exceeded, another log file is created. Each day, the oldest log files are replaced by the newer ones.

The log files are stored in:

IStream Document Manager_install_folder\Logs\daily log folders\log run subfolders\log files

The log folders and files are named in the following formats:

- daily log folder name – LOG_YYYY_MM_DD
- log run subfolders – RUN HH:MM:SS
- log files – RUNYYYYMMDDHHMMSS

Changing the Logging Level

You can change the logging level to control which level(s) of messages are displayed and logged.

Warning: This procedure involves editing the Windows registry and is therefore for system administrators only. We recommend that you back up your registry before proceeding. Also, shut down the server before making changes and restart after.

Note: The default setting is 3.

Method: To change the logging level

1. In the Windows registry, locate the following key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Oracle\IStream_Components\
Shared\Log\2.0
```

2. Set the **Logging Level** value. The following table indicates which level(s) of messages will be logged for each **Logging Level** value:

Logging Level	Info	Warning	Error	Critical
2	●	●	●	●
3		●	●	●
4			●	●
8				●
9	No messages logged.			

For example, a logging level of 4 will record **Error** and **Critical** messages only.

Logging Function

The Log Viewer works with the IStream Document Manager logging function called IT Logging. IT Logging is the actual process that logs all messages. This process runs automatically as a Windows service named `ITLogServer`.

The following applications use the logging function:

- Application launcher
- Assembler
- Assembler Link
- Assembly Server
- Assembler UI
- Author
- Compiler
- Customizer
- Helper object (works with Livelink)
- InfoConnector
- InfoSources
- PDF support
- Publisher Worker components
- IStreamXML InfoSource

The following applications use a proprietary logging system:

- IStream Document Manager Proxy (Java)
- IStream Document Manager Servlet (Java)

- LiveLink (Third party)
- Tomcat (Third party)

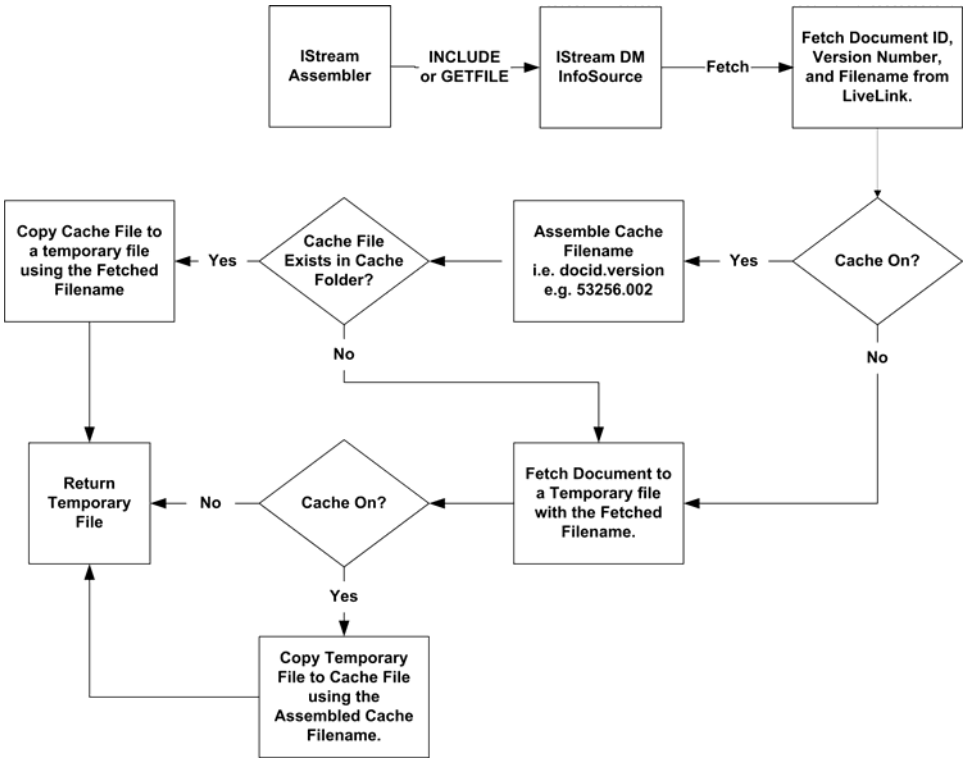
Document Caching

Document caching is a technique used to improve the performance of IStream Document Manager when IStream Assembler fetches documents from the IStream Document Manager server. It works by storing an image of the document on the cache folder after the document is fetched. The next time the same document is required, instead of fetching it from the IStream Document Manager server (which involves network traffic and downloading time), the cached file from the cache folder is returned.

Caching Documents for IStream Assembler

Only documents fetched by IStream Assembler using `INCLUDE` rule or the `GETFILE` function are cached. The document caching function does not apply to documents opened or saved through the IStream Repository, which is a component used to access documents by IStream applications such as IStream Author, IStream Customizer and IStream Promoter. The cached documents remain even when the source document has changed. Even if a document is cached, the IStream InfoSource still needs to connect to the IStream DM server to retrieve metadata about the document.

The following flowchart describes the flow when a document is cached and how the image file is retrieved from the caching folder with IStream Assembler using the `INCLUDE` rule or `GETFILE` function:



About the Cache Folder

All cache files are stored in the cache folder. Each cache should point to a different folder, which will be used exclusively for this purpose. When a document is fetched from the IStream Document Manager server, a file is created in the cache folder to hold the contents of the document. This file is used as a cache file for future reference to the same document.

Cache Folder Maintenance

To use document caching, cache folders have to be set up in InfoConnector.

Method: Set up a cache folder

The location and size of a cache folder (the maximum disk space dedicated to the cache) are set up in the **IStreamDM InfoSource Configuratio**n dialog box.

1. Launch InfoConnector.
2. Double-click the IStream DM InfoSource.
3. At the bottom of the **Configuration** dialog box, select the **Use Cache** radio button.
4. Enter the location and the maximum disk space to be dedicated to the cache, for example, 10MB.

Important: Two caches should not share the same folder. If they do, it can cause cached files to become out of sync with the IStream DM server and may cause incorrect cached files to be returned.

Method: To add a new file to a folder

When a document is fetched from the IStream Document Manager server, a temporary file imaging the document's content will be created in the cache folder. The Document ID and Version Number of the document will be used to assemble the file name (which will be in the format of `DocID.Version`) of the temp file, for example, `43256.002`.

Method: Add a new file to full folder

The cache folder is full when it contains the maximum cache size, which is the maximum amount of disk space dedicated to the cache defined during configuration. If a new file has to be added to the folder, then the least recently used file will be deleted to make space for the new file.

Another way to make more room for new files without deleting existing files is to to increase the disk space dedicated to the cache using the **IStream DM InfoSource Configuration** dialog box.

Method: Clean up the cache folder

You do not need to clean up the cache folder because the IStream DM InfoSource automatically does this. When the cache folder reaches its specified maximum size (see *Method:Set up a cache folder* on page 60) and a new file is added, the IStream DM InfoSource deletes the oldest modified file in the cache to make room for the new file.

About Cached Files

Files stored in the cache folder are the cached files. Each file contains the content of its corresponding, last-fetched IStream DM document. These files will be used instead of having to fetch documents from the IStream Document Manager server. However, you can only use the cached files when the cached files and respective IStream DM documents are synchronized; that is, they must have the same version number.

Synchronization of Cached Files

The content of a cached file and its corresponding IStream DM document should always be in sync if document caching has been enabled and has not stopped. However, other means of accessing a DMS document by not going through the IStream DM InfoSource component, such as adding a new version through the browser, will create a new version of a document. This will eventually cause the cached file to become out of sync, but then the version number will indicate that the document has a newer version. If this happens, IStream DM InfoSource will get the document again during the next fetching and update the cached file in the cache folder.

Document ID and Version Number

The Document ID and Version Number are two of the many attributes of a IStream DM document. The Document ID is an internal unique number assigned to a document and is maintained by the IStream Document Manager server. The Version Number reflects how many versions of a document have been created. Using this version number can verify that the cached file is synchronized with the latest version of a document on the IStream Document Manager server.

Limitations on Document Caching

Document caching is not equivalent to working offline. Every time IStream Assembler fetches a document, a connection to the IStream Document Manager server is still required.

Only documents referred by the `INCLUDE` rule or `GETFILE` function are cached. These include IStream Master documents, sections, Microsoft Word documents used through the `INCLUDE` rule and text files used through the `GETFILE` function. Any other documents are not cached.

Warning: Directly modifying the cached files is not recommended because it can cause problems synchronizing image files and IStream DM documents.

All computer sharing the same InfoSource configuration file (for example, `Local.idb`) will inherit the same caching behavior.

Plug-ins

IStream Document Manager has eight plug-ins which link to the appropriate web pages, related to the currently active model document in the Author. Links are available to the following pages:

- **Categories and Attributes** - access to the edit and view functions for categories and attributes, if you have sufficient privileges
- **Revision History** of the current model document
- **Audit Trail** - audit trail for the current model document
- **Properties** page for the current model document
- **Search** - takes you to the search page
- **Included in** - shows all model documents that include the current model **document** or section
- **Includes** - shows all sections the current model document includes
- **InfoSource dependencies** - shows all model documents referencing a specific InfoSource.

Audit Trail

IStream Document Manager tracks all of the actions taken against documents and other items. The system administrator (Admin user) chooses which events and items to audit. Users can view the audit trail on a per item basis by choosing the audit tab from the Info link on an item. The Admin user can view, query and purge the entire audit trail from the IStream Document Manager Administration page.

Method: Set audit trail options

1. Login as the administrator to the admin pages:

   server_name/*service_name*/`livelink.exe?func= Admin.index`
2. Choose Administer Event Auditing from the System Administration area.
3. Click Set Auditing interests.
4. Define which events you want to audit for specific items in the IStream Document Manager database.

Available Events

- create
- move
- delete
- copy
- rename
- reserve
- unreserve
- alias created
- permissions changed
- attributes changed
- version added
- version deleted
- version fetched
- purge
- revision created
- revision deleted
- release created
- release deleted
- CD ordered
- generation created
- login

Note: You have to restart the server for changes to take effect.

Querying the Audit Trail

The Admin user can query the audit trail by choosing the Query Audit log from the Administer Event Auditing area. This will allow the Admin to check on specific documents, types of objects, or users within specific date ranges.

Purging the Audit Trail

The Admin user can purge the entire audit log, or specific components of it based on object type, users, or date ranges. This is done from the Purge Audit Log link in the Administer Event Auditing area.

Promoting Documents

In IStream Document Manager, you can promote documents from one environment to another, typically from a development environment into a production environment, using IStream Promoter, the file promotion utility. This utility is described in detail in *IStream Promoter* on page 123.

The Model Document Development Process

The model document development process starts with creating a new model document and ends with the public issuance of that document. IStream Document Manager supports this process by defining a new model document workflow, and establishing how a model document gets initiated in development and eventually put into production, making it available to a Document Issuance process.

Using Environments

As describe above, the document process needs at least two environments, development and production, where documents can be generated. A document generated from a model document in the development environment may be different from a document generated from the same model in the production environment, because development may have implemented some revisions to the model and may be using different InfoSources.

There can be any number of environments a model document or set of sections must pass through before it can reach the production stage. Once the model document has passed through the required environments and testing, it can be promoted into production, and will then become visible in the product environment.

Identifying the Software Version

You can identify which version of an IStream product you have installed by clicking **Help** > **About**... to display the **About** box in Author, Assembler, or InfoConnector. This information is important for troubleshooting and is needed when you call Customer Support for assistance.

The version number is embedded in the executable and operational files, and displayed in the **About** box for Author, InfoConnector and Assembler (**Help** > **About**...).

Reserved Words

The following words, symbols, and operators may not be used as either field or table names for databases accessed by IStream Document Manager.

Reserved Words		
ABSOLUTE	ACTION	ADA
ADD	ALL	ALLOCATE
ALTER	AND	ANY
ARE	AS	ASC
ASSERTION	AT	AUTHORIZATION
AVG	BEGIN	BETWEEN
BIT	BIT_LENGTH	BOTH
BY	C	CASCADE
CASCADED	CASE	CAST
CATALOG	CATALOG_NAME	CHAR
CHAR_LENGTH	CHARACTER	CHARACTER_LENGTH
CHARACTER_SET_CATALOG	CHARACTER_SET_NAME	CHARACTER_SET_SCHEMA
	CHECK	CLASS_ORIGIN
CLOSE	COALESCE	COBOL
COLLATE	COLLATION	COLLATION_CATALOG
COLLATION_NAME	COLLATION_SCHEMA	COLUMN
COLUMN_NAME	COMMAND_FUNCTION	COMMIT
COMMITTED	CONDITION_NUMBER	CONNECT
CONNECTION	CONNECTION_NAME	CONSTRAINT
CONSTRAINT_CATALOG	CONSTRAINT_NAME	CONSTRAINT_SCHEMA
CONTINUE	CONVERT	CORRESPONDING
COUNT	CREATE	CROSS
CURRENT	CURRENT_DATE	CURRENT_TIME
CURRENT_TIMESTAMP	CURRENT_USER	CURSOR
CURSOR_NAME	DATA	DATE

Reserved Words		
DATETIME_INTERVAL _PRECISION	DATETIME_INTERVAL _CODE	DAY
DEALLOCATE	DEC	DECIMAL
DECLARE	DEFAULT	DEFERRABLE
DEFERRED	DELETE	DESC
DESCRIBE	DESCRIPTOR	DIAGNOSTICS
DICTIONARY	DISCONNECT	DISPLACEMENT
DISTINCT	DOMAIN	DOUBLE
DROP	DYNAMIC_FUNCTION	ELSE
END	END-EXEC	ESCAPE
EXCEPT	EXCEPTION	EXEC
EXECUTE	EXISTS	EXTERNAL
EXTRACT	FALSE	FETCH
FIRST	FLOAT	FN
FOR	FOREIGN	FORTRAN
FOUND	FROM	FULL
GET	GLOBAL	GO
GOTO	GRANT	GROUP
HAVING	HOUR	IDENTITY
IGNORE	IMMEDIATE	IN
INCLUDE	INDEX	INDICATOR
INITIALLY	INNER	INPUT
INSENSITIVE	INSERT	INT
INTEGER	INTERSECT	INTERVAL
INTO	IS	ISOLATION
JOIN	KEY	LANGUAGE
LAST	LEADING	LEFT
LENGTH	LEVEL	LIKE
LOCAL	LOWER	MATCH
MAX	MESSAGE_LENGTH	MESSAGE_OCTET_LENGTH
MESSAGE_TEXT	MIN	MINUTE
MODULE	MONTH	MORE

Reserved Words		
MUMPS	NAME	NAMES
NATIONAL	NATURAL	NCHAR
NEXT	NO	NONE
NOT	NULL	NULLABLE
NULLIF	NUMBER	NUMERIC
OCTET_LENGTH	OF	ON
ONLY	OPEN	OPTION
OR	ORDER	OUTER
OUTPUT	OVERLAPS	PAD
PARTIAL	PASCAL	PLI
POSITION	PRECISION	PREPARE
PRESERVE	PRIMARY	PRIOR
PRIVILEGES	PROCEDURE	PUBLIC
READ	REAL	REFERENCES
RESTRICT	RETURNED_LENGTH	RETURNED_OCTET_LENGTH
RETURNED_SQLSTATE	REVOKE	RIGHT
ROLLBACK	ROWS	RTRIM
SCALE	SCHEMA	SCHEMA_NAME
SCROLL	SECOND	SECTION
SELECT	SERIALIZABLE	SERVER_NAME
SESSION	SESSION_USER	SEQUENCE
SET	SHUTDOWN	SIZE
SMALLINT	SOME	SPACE
SQL	SQLAC	SQLCODE
SQLERROR	SQLSTATE	SQLWARNING
SUBCLASS_ORIGIN	SUBSTRING	SUM
SYSTEM	SYSTEM_USER	TABLE
TABLE_NAME	TABLESPACE	TEMPORARY
THEN	TIME	TIMESTAMP
TIMEZONE_HOUR	TIMEZONE_MINUTE	TO
TRAILING	TRANSACTION	TRANSLATE
TRANSLATION	TRIM	TRUE

Reserved Words		
TYPE	UCASE	UNCOMMITTED
UNION	UNIQUE	UNKNOWN
UNNAMED	UPDATE	UPPER
USAGE	USER	USING
VALUE	VALUES	VARCHAR
VARCHAR2	VARYING	VIEW
WHEN	WHENEVER	WHERE
WITH	WORK	WRITE
YEAR	ZONE	::=
|	||	"
%	&	'
(	)	*
,	-	.
/	:	;
<	>	=
?	{	}
[	]	<>
>=	<=	...

Edit and Author Functions

The **Edit** and **Author** functions allow IStream Document Manager users to launch editing sessions on sections from the DMS. The **Edit** function opens the section in Microsoft Word for editing. Users can use the Author Add-in for Microsoft Word to access some important Author features, such as the compilation and generation functions.The **Author** command opens the section in IStream Author for editing, and makes all Author functionality available to the user editing the section.

The **Edit** function is also available on IStream documents stored on the DMS. The document opens in Microsoft Word for editing, and you can use the Author Add-in for Microsoft Word to regenerate it.

The *IStream Document Manager Online Help* provides instructions on how to use the these functions.

The *IStream Author Online Help* provides instructions on how to prepare model documents for use with the Edit function.

The configuration parameters are found in the `Opentext.ini` file. After modifying these parameters, be sure to shut down and restart the Livelink (IStream Document Manager) services.

Opentext.ini

In the **Name** column of the table below, the section of the `opentext.ini` file where the parameter is found is in square brackets [].

Name	Description	Example
`Calligo_locServlet [general]`	The location of the IStream Document Manager servlet context.	*:8080/calligo* (In this example, the Tomcat server must be set up for port 8080.
`ISCName [general]`	Specifies the InfoSource name used by IStream Document Manager	IStreamDM

The IStream Document Manager Servlet

The IStream Document Manager Servlet provides the connection between IStream Document Manager and the Assembly Server. The IStream Document Manager Servlet contains two Java servlets:

- **InteractiveServlet** – used when IStream Publisher is not installed
- **NonInteractiveServlet** – used when IStream Publisher is installed.

In IStream Document Manager, only one servlet can be enabled at any one time. The servlets are enabled by setting the "mode" parameter in the `CalligoServlet.configure` file. See *CalligoServlet.configure* on page 74 for configuration instructions.

InteractiveServlet provides document generation when IStream Publisher is not installed as part of the IStream Document Manager system. It is the default servlet for Web-based document generation.

NonInteractiveServlet uses IStream Publisher to provide document generation functions. NonInteractiveServlet should only be enabled when IStream Publisher is installed. Due to the nature of IStream Publisher, this servlet cannot provide real-time document generation log messages as is seen when using the InteractiveServlet.

Servlet Dependency

The IStream Document Manager servlet is a Java-based servlet application that depends on the following software modules:

- Jakarta Tomcat: used as the Java servlet engine.
- IStream Publisher: works with NonInteractiveServlet only
- Assembly Server: works with InteractiveServlet only
- IStream Document Manager Proxy: provides document compilation and PDF rendering.
- FTP Server: provides document repository for both IStream Publisher/ NonInteractiveServlet combination and InteractiveServlet.

The following table displays the dependency between different IStream Document Manager software modules and servlets to produce the various output types in the left column. The bullets indicate where a dependency exists.

IStream Document Manager Features (includes Document Regeneration)	IStream Document Manager Modules Dependency					
	IStream Publisher	Non-Interactive Servlet	Interactive Servlet	FTP Server	IStream Document Manager Proxy	Assembly Server
	Dependencies When Using Non-Interactive Servlet					
Document Generation CLG	●	●		●		
Document Generation DOC	●	●		●		
Document Generation PDF	●	●		●		
Document Compilation with IStream Publisher	●	●		●	●	
	Dependencies When Using Interactive Servlet					
Document Generation CLG			●			●
Document Generation DOC			●			●
Document Generation PDF			●	●	●	●
Document Compilation without IStream Publisher			●	●	●	

IStream Document Manager Servlet Configuration

Follow these instructions to configure the IStream Document Manager Servlet on a Microsoft Windows system. This servlet is installed during the normal installation process, and some configuration is done during installation.

Refer to *Install the IStream Modules and Servlet* in the *IStream Document Manager Install Guide* for installation and configuration notes.

Three files contain additional IStream Document Manager Servlet configuration information:

- *CalligoServlet.configure*
- *InteractiveServlet.configure*
- *NonInteractiveServlet.configure*

These files are found in the directory:

```
<Tomcat root>\webapps\Calligo\conf
```

CalligoServlet.configure

Name	Description	Example
mode=normal	normal | edelivery Determines if IStream Publisher will be used or not. By default normal is set and IStream Publisher will not be used. To use IStream Publisher, set this property to `edelivery`.	edelivery

InteractiveServlet.configure

Parameter Name	Description
InteractiveServlet.action.generate	Generate action string used in action.xml. Example: igenerate.do
InteractiveServlet.view.error	View error action forward name used in action.xml. Example: error
InteractiveServlet.view.fail	View failure action forward name used in action.xml. Example: fail
InteractiveServlet.view.succcess	View success action forward name used in action.xml. Example: success
InteractiveServlet.view.generate	View generation progress action forward name used in action.xml. Example: showGenerate
InteractiveServlet.server.address	The Assembly Server name or IP address. This field can be empty.
InteractiveServlet.server.port	Assembly Server TCP port number. This field can be empty. 4300 is the default TCP port number used by Assembly Server. This field is used only when the port number is not supplied by the DMS server.

Parameter Name	Description
InteractiveServlet.document.localpath	The local file system path that is also used as the FTP site in IStream Document Manager. This field cannot be empty. For IIS FTP site, this is the logical file system directory that is also used as the FTP site. For example: `C:\\inetpub\\ftproot\\` **Note:** This path must end with the platform dependent directory separator symbol. For Windows operating systems, each separator symbol must be specified twice as shown.
CalligoRMI.server.hostname	IStream Document Manager Proxy name or IP address. This field cannot be empty. IStream Document Manager Proxy provides the ability to: • compile a model document and check it in to a DMS server • render a Microsoft WORD document to Acrobat PDF format and check it in to a DMS server
InteractiveServlet.config.SrcISC	Default IStream Document Manager ISC name used as document source path. **Example**: IStreamDM **Note:** On some operating systems, this name may be case-sensitive.
InteractiveServlet.config.DstISC	Default IStream Document Manager InfoSource name used as document destination path. **Example**: IStreamDM **Note:** On some operating systems, this name may be case-sensitive.
InteractiveServlet.config.FtpServer	FTP server name or IP address. This FTP server provides a dedicated FTP site for IStream Document Manager internal use. This field cannot be empty.
InteractiveServlet.config.FtpPort	The default FTP server port number: usually this is 21.

Parameter Name	Description
InteractiveServlet.config.FtpTempPath	The folder path under the dedicated IStream Document Manager FTP site. For IIS FTP site, this corresponds to the virtual directory name. For example: /temp/ **Notes:** All directory paths must end with the directory separator symbol for the local operating system. You must specify a value for this parameter or PDF generation using Assembly Server will not work.
InteractiveServlet.config.FtpUserId	The proxy user ID for the dedicated IStream Document Manager FTP site. For IIS FTP site, this can be user anonymous. This field cannot be empty.
InteractiveServlet.config.FtpPassword	The password for the proxy user defined in the FtpUserId field. If there is no password, leave this field empty.

NonInteractiveServlet.configure

Only the significant properties are listed.

Name	Description	Example
CalligoSend.config.QueueName	Name of the queue for sending generation requests. This should match the name of the Submission Queue in the Publisher configuration.	request
CalligoSend.config.DefaultSrcInfoSource	Name of the InfoSource to use if one is not specified for the model document.	IStreamDM
CalligoSend.config.DefaultDstInfoSource	Name of the InfoSource to use if one is not specified for the generated document.	IStreamDM
CalligoReply.config.QueueName	Name of the queue dedicated for this server. Must be unique for each DMS server.	response
CalligoReply.config.msTimeout	The maximum number of milliseconds to wait for a reply.	60000

Name	Description	Example
CalligoRMI.server.hostname	Server name where the IStream Document Manager Proxy server is installed and to be invoked.	RMIServer
Calligo.DebugLevel	Detail level of debug messages generated into the Tomcat server console. 0 – none 1 – minimal trace messages 2 – print request and reply message contents as well as other trace messages	0
CalligoSendFactory.setup.setHostName	Host Server Name	server.IStream.com
CalligoSendFactory.setup.setQueueManager	Queue Manager	QM_username
CalligoSend.name	Request Queue	submit
CalligoReplyFactory.setup.setHostName	Host Server	server.IStream.com
CalligoReplyFactory.setup.setQueueManager	Queue Manager	QM_username
CalligoReply.name	Reply Queue	response
CalligoSend.config.FtpServer	Used for PDF support. Determines the FTP server to be used for intermediate files.	ftp.company.com
CalligoSend.config.FtpPort	Used for PDF support. Determines the port that the FTP Server is access through. This value must be 21.	21
CalligoSend.config.FtpTempPath	Used for PDF support. Determines the subdirectory from the root of the FTP server that should be used to hold temporary files.	temp
CalligoSend.config.UserId	Used for PDF support. The user id required to log on to the FTP server.	anonymous

Name	Description	Example
CalligoSend.config.FtpPassword	Used for PDF support. The password required to log on to the FTP server. **Note:** You must have a password for anonymous as well depending on your configuration	
CalligoSend.config.PdfDriver	Used for PDF support. The name of the printer installed on the IStream Publisher server to be used for writing PDF files.	
CalligoResubFactory…	A set of parameters whose names start with "CalligoResubFactory". These are used for IStream Publisher aggregate requests. Currently, aggregate requests are only used to create PDF files. The parameters are identical to those used in CallligoReplyFactory settings. It is likely that the values will be the same with the exception of the CalligoResub.name parameter which will likely contain a queue name similar to **service** instead of **response**.	

IStream Document Manager Servlet Interface Parameters

The following tables provide information about the IStream Document Manager Servlet Interface parameters:

InteractiveServlet

NonInteractiveServlet

InteractiveServlet

Parameter name	Description
State	Servlet task state CalligoServlet client must provide "st_setup" as the first state of the servlet
ServerAddress	Assembly Server name or IP address
ServerPort	Assembly Server TCP connection port number
UserID	IStream Document Manager server - user login ID
UserPswd	IStream Document Manager server - user login password
ProxyID	IStream Document Manager server - proxy user login ID
ProxyPswd	IStream Document Manager server - proxy user login password.
ModelName	IStream model document UISR path
Variables	IStream model document variables list Not in use.
GenMode	IStream Assembler document generation mode Not in use
DestType	Generated document type “calligo” = CLG generation "word" = DOC generation "pdf" = PDF generation
DocumentName	Generated document save path (UISR).
DMSBaseURL	IStream Document Manager base URL. Example: `http://dmscomputer/IStream/livelink.exe`
MessagingLevel	Interactive messaging level
nextUrlSuccess	the URL that the IStream Document Manager servlet will go to when it finishes successfully completes its task
nextUrlFail	the URL that the IStream Document Manager Servlet will go to when it fails its task

Parameter name	Description
DBListener	IStream Document Manager login information. Syntax: `connection;computername:TCPportnumber`
keydata	List of model document key data Syntax: ### is data delimiter Each complete key data contains 4 fields: `variableName###dataType###description###value###`

NonInteractiveServlet

Parameter name	Description
SrcType	IStream document type. Possible values: "cms" - model document “clg” - IStream document
SourceName	Name of a Model document or the IStream document. If this parameter is not provided then: • If SrcType is "cms", then SourceName is taken from ModelName. • If SrcType is "clg", then SourceName is taken from DocumentName. IStream: “Models\Model1\Model1” for new document. IStream: “Generated\Doc1.clg” to regenerate.
DestType	Generated document type. "calligo" = CLG generation "word" = DOC generation "pdf" = PDF generation
DocumentName	Generated document save path (UISR).
DMSBaseURL	IStream Document Manager base URL. Example: `http://dmscomputer/IStream/livelink.exe`
nextUrlSuccess	IStream Document Manager servlet will go to this URL when successfully finishes.
nextUrlFail	IStream Document Manager will go to this URL when it fails to finish.
DBListener	IStream Document Manager login information. Syntax: `connection;computer_name:TCPportnumber`
keydata	List of model document key data. Syntax: ### is data delimiter Each complete key data contains 4 fields: variableName###dataType###description###value###
template	IStream Document Manager UISR template path. Used by IStream Document Manager Proxy compile service only.

Parameter name	Description
user	User ID. Used by IStream Document Manager Proxy compile service only.
password	User password (unencrypted). Used by IStream Document Manager Proxy compile service only.

Generating With IStream Publisher

IStream Publisher, IStream Document Manager, Tomcat server, and the IStream Document Manager Servlet must be installed properly as outlined previously.

The example(s) in this section assumes that the Tomcat server was installed on the same server as IIS driving IStream Document Manager in Standalone mode (all using default port assignments).

CalligoServlet.Generate

The following parameters are supported by this servlet action.

Name	Description	Example
ModelName	(Optional) The full path name of the model document when generating a new document. For regeneration this parameter is ignored and should be an empty string. This parameter is provided for compatibility with IStream DM. If SrcType is “cms”, then either ModelName or SourceName must be provided.	`IStream:"Models\Model1\Model1" for new document.`
SrcType	["cms" | "clg"]. Default: cms. Identifies the type of the source file provided in SourceName. If SourceName is not provided (when called from IStream Document Manager), the SrcType determines whether to use the ModelName or the DocumentName as the SourceName.	`cms`

Name	Description	Example
SourceName	This parameter is optional for compatibility with IStream Document Manager. It is the name of a Model document or the IStream document. If this parameter is not provided then SrcType determines where to obtain the source file name from as follows: If SrcType is "cms", then SourceName is taken from ModelName. If SrcType is "clg", then SourceName is taken from DocumentName.	`IStream:"Models\Model1\Model1" for new document.` `IStream:"Generated\Doc1.clg" to regenerate.`
DestType	Specifies the generated document file type. It must be one of {“IStream” | “word” | “pdf”}, to generate a “clg”, Microsoft Word, or PDF file, respectively. The default is “calligo”.	`calligo`
DocumentName	The full path name of the destination IStream document. This should contain the name of the document to be (re)generated. For regeneration the DocumentName may be the same as the ModelName. It is also possible to regenerate a clg document and create a new clg document under a different name. This parameter can also be used to regenerate a clg document to create a Microsoft Word or a PDF document.	`IStream:"Generated\OutputDoc1"` `IStream:"Generated\Output.pdf"`
UserID	User id for the generation.	
UserPswd	Password for the generation.	
ProxyID	Proxy User id for the generation. If this parameter is not specified, then the value of *User* is used as the ProxyID.	
ProxyPswd	Proxy password for the generation. If *ProxyID* is not specified, then the value of *Password* is used.	
nextUrlSuccess	URL of the page to forward to if the generation succeeds. The URL should contain the URL address along with the query string needed by the page. See *NextURLSuccess example* on page 84 for an example of how this is used.	
nextUrlFail	URL of the page to forward to if the generation fails.	

Name	Description	Example
DBListener	String of the format: "<DMS-Connection>;<DMS-Server>:<DMS-port>". Each of the 3 <…> values can be obtained from the ISC configuration.	`IStreamdb/dms2:2099`
keydata	List of key data elements to pass to the document generator. The format of each keydata element has the form: <kd-name>###<kd-type>###<kd-description>###<kd-value> Multiple keydata elements has the form: <kd-element1>###<kd-element2>###<kd-element3>… <kd-type> must be one of the following: string, number, logical, date <kd-value> for logical types must be "true" or "false" in lower or upper case. <kd-value> for date types must be in the format "mm/dd/yyyy".	

NextURLSuccess example

```
http://servername:80/IStream/livelink.exe?func=ll&
objtype=7771&objAction=doc.View&nodePath=Generated\docname
```

Invoking CalligoServlet.Generate (No key data) – HTML Example

```
<html>
<head>
<title>
Generate
</title>
</head>
<body>

<form action="/servlet/calligoservlet.Generate" method="post">

<TABLE CELLPADDING="1" CELLSPACING="1" BORDER="0" ALIGN="CENTER">
<TR>
  <TD> </TD>
  <TD NOWRAP ALIGN="LEFT"><FONT SIZE="4"><B><U>Generation Requester</FONT></U></
B></TD>
</TR>

<TR><TD> </TD></TR>
```

```
<TR>
  <TD NOWRAP ALIGN="RIGHT">Username:</TD>
  <TD ALIGN="LEFT"><INPUT TYPE="TEXT" NAME="UserID" SIZE="22"></TD>
</TR>

<TR>
  <TD NOWRAP ALIGN="RIGHT">Password:</TD>
  <TD ALIGN=LEFT><INPUT TYPE="PASSWORD" NAME="UserPswd" SIZE="22"></TD>
</TR>

<TR>
  <TD NOWRAP ALIGN="RIGHT">Proxy Username:</TD>
  <TD ALIGN="LEFT"><INPUT TYPE="TEXT" NAME="ProxyID" SIZE="22"></TD>
</TR>
<TR>
  <TD NOWRAP ALIGN="RIGHT">Proxy Password:</TD>
  <TD ALIGN=LEFT><INPUT TYPE="PASSWORD" NAME="ProxyPswd" SIZE="22"></TD>
</TR>

<TR><TD> </TD></TR>

<TR>
  <TD NOWRAP ALIGN="RIGHT">Source Model Path:</TD>
  <TD ALIGN=LEFT><INPUT TYPE="text" NAME=" ModelName " SIZE="40"></TD>
</TR>

<TR>
  <TD NOWRAP ALIGN="RIGHT">Destination Path:</TD>
  <TD ALIGN=LEFT><INPUT TYPE="text" NAME="DocumentName" SIZE="40"></TD>
</TR>

<TR>
  <TD NOWRAP ALIGN="RIGHT">Destination Type:</TD>
  <TD ALIGN=LEFT>
    <SELECT  NAME="DestType" SIZE="1">
    <OPTION SELECTED VALUE="calligo">IStream document
    <OPTION VALUE="word">Microsoft Word Document
    </SELECT>
  </TD>
</TR>

<TR><TD> </TD></TR>

<TR>
  <TD NOWRAP COLSPAN="1" ALIGN="RIGHT">Next URL on Success:</TD>
  <TD ALIGN=LEFT><INPUT TYPE="text" NAME="nextUrlSuccess" SIZE="60"></TD>
</TR>

<TR>
  <TD NOWRAP COLSPAN="1" ALIGN="RIGHT">Next URL on Fail:</TD>
  <TD ALIGN=LEFT><INPUT TYPE="text" NAME="nextUrlFail" SIZE="60"></TD>
</TR>

<TR>
  <TD NOWRAP COLSPAN="1" ALIGN="RIGHT">Database Name/Port:</TD>
  <TD ALIGN=LEFT><INPUT TYPE="TEXT" NAME="DBListener" VALUE="Livelinkdb/
server:2099" SIZE="40">
</TR>
```

```
<TR><TD> </TD></TR>
```

```
<TR>
  <TD ALIGN="RIGHT"><INPUT type="SUBMIT" name="Submit"
value="Generate"></TD>
  <TD ALIGN="LEFT"><INPUT type="RESET" name="Reset"
value="Clear"></TD>
</TR>
</TABLE>

</form>
</body>
</html>
```

The sample code sends a post action to the Generate servlet action to generate a document. The **bold** text in the example lists all the parameters accepted by the IStream Generate Servlet.

Load Balancing

To cluster IStream Document Manager Servers for the purpose of load balancing, an IP router with the ability to persist session connections is required. Such an IP router keeps track of each browser session and routes information to the same server for as long as the session is active.

The persistent session function is required because of the way the IStream Document Manager Servlet caches data to the HTTP session within the servlet.

Note: This is only required when using the Servlet, which is only used when Assembly Server is deployed and active. The TCP load balancer will fail if the IStream Document Manager service is the point of failure, and not the HTTP server.

PDF Rendering Properties

This section describes how to adjust the default settings used when authors render documents to PDF.

Automatic TOC Update Registry Setting

When rendering a document into a PDF, the table of contents in the document is updated automatically. This automatic update function is installed as active when you install Author. If an Author user is using an existing macro to perform this task and they want to continue to use only the macro, the automatic update function can be disabled. Use the steps in this section to change the registry settings of a Author user's computer to disable the automatic update function.

Method: Disable the automatic TOC update when rendering a PDF

1. In the Registry Editor, navigate to

   `HKEY_LOCAL_MACHINE\SOFTWARE\Oracle\IStream_Components\PDFRendering`

2. Select the setting named **UpdateTOC**, choose **Modify**, change the value data to 0 to disable the automatic TOC update function.

3. Close the Registry Editor.

PDF Minimum Margins

You can adjust the default sizes of the minimum horizontal (top and bottom) and vertical (left and right) margins that appear in the PDF files generated by authors.

Method: Adjust PDF minimum margin sizes

1. In the Registry Editor, located the following key:

   `HKEY_LOCAL_MACHINE\SOFTWARE\Oracle\IStream_Components\PDFRendering`

2. Within this key, create the following `DWORD` values:

   - `HorizontalMargin`
   - `VerticalMargin`

3. Enter the values for these keys. The units for these keys are .1 mm. A value of 10, therefore, would produce a margin of 1 mm. The default value is 60 (6 mm), which is also the maximum value.

Configuring PDF Font Embedding Options

The following procedure describes the various options for embedding fonts within PDFs and how to configure these options.

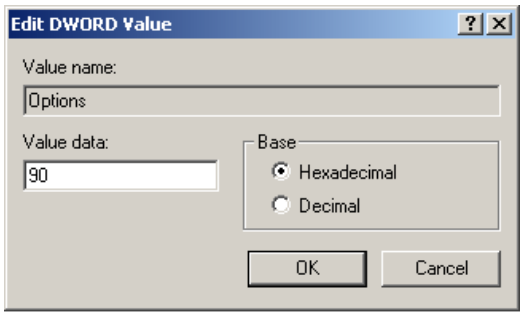
Method: Enable font embedding

Warning: This procedure involves editing the Windows registry editor and is therefore for system administrators only. We recommend that you back up your registry before proceeding.

1. On the Author Workstation and Assembly Server systems, locate the following registry key:

   `HKEY_LOCAL_MACHINE\SOFTWARE\Oracle\IStream_Components\PDFRendering`

2. Open the `Options` DWORD value.



The **Value data** setting controls the font embedding options, as follows:

Value data	Partial font embedding	Multilingual support	Full font embedding
10	●		
90	●	●	
80		●	
210			●

By entering a specific **Value data,** you select one or more of these options as indicated by this table. For example, if you enter `10` in the **Value data** field, you will enable partial font embedding.

The default value is `90`, which selects the partial font embedding and multilingual support options.

Note: You can only select the values indicated in this table. This means that some options are mutually exclusive.

The following list describes each of these options in detail:

- **Partial font embedding** – If you select this option, then only the non-standard, non-licensed, non-simulated fonts used in the source document are embedded in the PDFs. The embedded fonts contain only characters that are in the source document.
- **Multilingual support** – If you select this option, then support is added to the PDF for international character sets. This allows the PDFs to contain unicode characters.
- **Full Font embedding** – If you select this option, then the *entire* character set of fonts that is used in the source document is embedded in the PDF. Again, these fonts must be non-standard, non-licensed, and non-simulated.

Important: The partial or full font embedding options increases the PDF file size. Therefore, you may choose not to select these options if you are sure that others reading your PDFs have the fonts used in the document on their systems.

For more options, see pages 22 to 23 of the *Developer's Manual of Amyuni Document Converter* (). The document is in:

IStream Document Manager folder\3rd Party Components\PDFDriver\Common Driver Interface 250.pdf.

However, note that not all of the options described in this PDF are supported with or apply to IStream Document Manager.

Configuring File Options in the Zip & Download and Print Features

The `[options]` section in `opentext.ini` includes the following settings that control how IStream Document Manager will handle files and folders for the **Zip & Download** and **Print** features.

- **CalligoOriginalExtension** – If `true`, IStream Document Manager will not add an extension to the file name if there already is one
- **CalligoNoDavExtensions** – If `true`, IStream Document Manager will not change the downloaded file and folder name. This is convenient if the user wants to download the file or folder as it appears in the DMS.

`Important:` Before editing `opentext.ini`, stop the DMS LiveLink server if it is running.

Chapter 6

XML InfoSources

An XML InfoSource is an InfoSource that uses XML as a source for data.

This chapter describes:

- *XML InfoSources* on page 94
- *Structured XML InfoSource* on page 96
- *XPath InfoSource* on page 103
- *Adding XML InfoSources* on page 115
- *Coding Model Documents for XML InfoSources* on page 116
- *Author Wizards and XML* on page 120

XML InfoSources

The Assembly Engine uses the XML InfoSource during document generation to retrieve data from the XML file.

XML InfoSources are:

- driven by XML, which is becoming the standard for data exchange and document creation processes
- faster than document generation from a database, because pre-extracted data are read once, and stored in local memory
- an effective solution to the problem of legacy data being stored on mainframes or third party systems
- useful in a network environment, because you can send documents to multiple printers and job computers
- a scalable solution that allows you to add worker systems and printers as production volumes increase

You can use an XML InfoSource:

- the *structured way* that XML data are organized according to the IStream structure requirements, that is, using IStream pre-defined tag names such as `row`); an XML InfoSource used this way is called the *structured XML InfoSource*

  -or-

- the *XPath way*, where the XML data structure is arbitrary, and the XML data are mapped to an IStream Query statement using XPath; an XML InfoSource used this way is called the *XPath InfoSource*

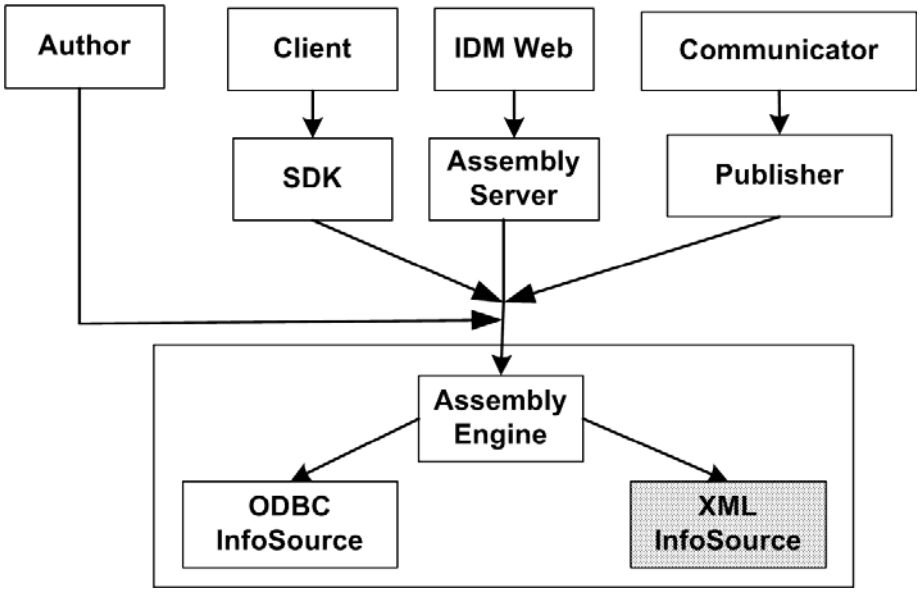
The structured XML InfoSource was primarily designed for processing XML bulk data for Calligo 5.4 Dispatcher. This type of XML InfoSource has been retained in the IStream product suite for backward compatibility. However, you should use the XPath InfoSource for any new applications.

Note: XML InfoSources are called IStreamXML InfoSources in the InfoConnector application.

Architectural Overview

This section gives an architectural overview on the place of the XML InfoSource in the system and the business purpose it can serve.

Many IStream solutions are built on top of IStream Assembler which generates documents in IStream. ODBC and XML InfoSources are available for IStream Assembler.



The XML InfoSource can be configured to use (or not use) XPath.

With XPath, the InfoSource can retrieve data from canonical XML files.

If you are using the structured XML InfoSource, this Infosource can retrieve data from an XML data file which conforms to the IStream structure.

Structured XML InfoSource

This section describes how to prepare a structured XML InfoSource.

Preparing the XML File

In structured XML InfoSources, data for document generation are retrieved from an XML file containing data extracts from a database or another data source, such as a mainframe or legacy system. You prepare the XML file before document generation begins.

This section provides some guidelines to follow while preparing the XML file for batch mode or interactive mode use, describes the structure of the file and how to represent data in it.

Structuring the XML File

You should structure your XML file according to whether you are using it for batch or interactive processing. This section provides guidelines for structuring an XML file for both types.

Ensure that your XML code is well-formed. All the data provided in the XML file for document assembly must be formatted according to the syntax described in this chapter. The XML InfoSource does not validate the completeness or correctness of the data structure in the XML file.

Structuring an XML File for Batch Mode Processing

The XML file prepared for batch mode processing should contain the data extracts required to generate each IStream Document in the batch. All the data required for the generation of a particular document are grouped in one block, with a unique `job ID` number. The XML file can have multiple blocks of data to generate of many documents.

The following guidelines help you structure an XML file to be used in batch mode:

- Place the tags describing all data in the batch between these tags:

  ```
  <batch>
  ...
  </batch>
  ```

  where `batch` is the reserved tag name. There can be only one `batch` tag in one XML file, and this tag must be the root tag in the file.

- Place the data required for one document generation between these tags:

  ```
  <job id="xxxxxx">
  ...
  </job>
  ```

where `xxxxxx` is the job ID number, which must be a positive integer. The number of digits depends on the total number of batches. The tag name `job` and its attribute name `id` are reserved names. The job tag must be the immediate child tag of batch tag.

- To ensure the best performance, places the blocks of data in the XML file in order of job ID, from lowest to highest.
- When an IStream model document processes the batch mode XML data, the IStream `Query` statement within the model document must be coded properly to ensure that during generation, the job ID number passed to the `Query` statement follows the order of the job ID numbers in the XML data. If the job ID passed to the Query statement is hard-coded with a constant number or does not follow the order of job IDs in the XML data, then the IStream Assembly Engine may not generate the same model repeatedly more than twenty times.

Structuring an XML File for Interactive Mode Processing

The XML file for interactive mode processing should contain the data extracts required to generate a single IStream Document.

Place all data between this single pair of tags:

```
<interactive>
...
</interactive>
```

Note: `interactive` is the reserved tag name and must be the root tag in the XML file. Do not use the <`job id`> tags because these are used for batch processing.

Representing Data in the XML File

Data required to generate a document can be *normal (scalar) variable*, or *recordset*. This section describes how to represent both types of data in the XML file.

Representing Normal (Scalar) Variable Data

Define normal (scalar) variable data in the XML file as follows:

```
<varname type="datatypename">Value</varname>
```

where:

- `varname` is both the tag name and the variable name interpreted by the IStream Assembly Engine. Do not use the same tag name twice in the same job block of data.
- The attribute `type` identifies the data type and is a reserved attribute name. Valid values for datatypename are:
  - `double` - corresponding to the IStream `Numeric` type
  - `string` - corresponding to the IStream `Character` type

- `date` - corresponding to the IStream `Date` type
- `boolean` - corresponding to the IStream `Logical` type.

Note: The values for the date type must follow the IStream character date representation format. By default, the date format in IStream is `MM/DD/YYYY`.

Representing Recordset Data

Represent recordset data in the XML file using the following structure:

```
<RecordsetName>
   <FieldName type="array">
      <row type="datatypename">Value1</row>
      ...
      <row type="datatypename">ValueN</row>
   </FieldName>
   ...
</RecordsetName>
```

where:

- `RecordsetName` is both the tag name and the name of a recordset interpreted by the IStream Assembly Engine. Do not use the same `RecordsetName` tag twice in the same job block of data.
- `FieldName` is both the tag name and the name of a variable corresponding to the recordset's field name, as interpreted by the IStream Assembly Engine. Do not use the same `FieldName` tag in the same recordset tag.
- The attribute `type` identifies the data type and is a reserved attribute name. Its only valid value for the `FieldName` tag is `array`.

Recordset Field Values

All the possible values that a particular field can have within a recordset must be specified between the `FieldName` tags, as follows:

```
<row type="datatypename">Value</row>
```

where:

- `row` is a reserved tag name
- The Attribute `type` identifies the data type, and is a reserved attribute name. Valid values for `datatypename` are the same as for the normal (scalar) variable described previously.

Note: The number of row tags must be the same for all fields within a recordset, and all row tags must have same `datatypename` values for the same field.

Common Problems with Representing Data in an XML File

Representing data in an XML file can cause various problems which require creative solutions. Solutions to the some of the more common problems are described in this section.

Identical Field Names within the Same Recordset

You want to query multiple database tables but they contain fields with identical names within the same recordset. You can solve this problem by:

- Using different aliases in the XML file to represent the identical field names. Use the alias name (rather the original field name) in the model document's code.
- Using composite field names in the XML file to represent the field, for example, `Tablename.Fieldname`. Use the composite field name in the IStream Model document.

Undefined Values in the XML File

To leave the value of a variable or a recordset field undefined in the XML file, code the XML file as follows:

```
<varname type="datatypename"></varname>
```

-or-

```
<row type="datatypename"></row>
```

When the value is undefined in the XML file, the IStream Assembly Engine will interpret the code as follows

Data	Interpretation of undefined value
Double	0
String	Empty
Date	Empty
Boolean	False (F)

Batch Mode Processing – Example XML File

This section contains an example of an XML file which follows the guidelines outlined in *Structuring the XML File* on page 96. An explanation of this example follows. For an example of model document code that uses this file as an InfoSource, see *Example Model Document Code* on page 119.

Note: For the purpose of this example, each line is numbered. Do not include these numbers in your XML file.

```
1 <?xml version="1.0" encoding="ISO-8859-1" ?>
2 <batch>
3  <job id="66601">
4  <insured type="string">Full Time Employees</insured>
5  <eff_date type="date">12/12/2012</eff_date>
6  <pol_num type="double">66601</pol_num>
7  <max_ben type="double">25000</max_ben>
8  <FULL>
9    <benefit type="array">
10    <row type="string">1</row>
11    <row type="string">2</row>
12    <row type="string">3</row>
13   </benefit>
14   <deduc_per_fam type="array">
15    <row type="string">25.00</row>
16    <row type="string">50.00</row>
17    <row type="string">100.00</row>
18   </deduc_per_fam>
19   <pay_date type="array">
20    <row type="date">01/01/1999</row>
21    <row type="date">03/31/1999</row>
22    <row type="date">06/30/1999</row>
23   </pay_date>
24   <reimbursement type="array">
25    <row type="double">80</row>
26    <row type="double">100</row>
27    <row type="double">120</row>
28   </reimbursement>
29 </FULL>
30 <BENEFITS>
31   <CODE type="array">
32    <row type="double">1</row>
33    <row type="double">2</row>
34    <row type="double">3</row>
35    <row type="double">4</row>
36    <row type="double">5</row>
37   </CODE>
38   <RIDER type="array">
39    <row type="string">Formulary Drug Plan</row>
40    <row type="string">Hospital: Ward to Semi-Private</row>
41    <row type="string">Supplemental Health Care</row>
```

```
42    <row type="string">Vision Care</row>
43    <row type="string">Intensive Care Rider</row>
44   </RIDER>
45 </BENEFITS>
46  </job>
47  . . .
48 </batch>
```

Explanation

- Line 1 indicates that this is an XML file built according to the conventions of XML version 1.0.
- Lines 2 and 48 define the opening and closing of the root tag named `batch`.
- Lines 3 and 46 define the opening and closing tags for a block of data corresponding to a job with a job ID number of 66601.
- Lines 4-7 define four variables, including their names, types, and values.
- Lines 8 and 29 define the opening and closing tags for the group of data representing the recordset named `FULL`.
- Note that in lines 9-28, the recordset `FULL` is presented as a group of one-dimensional arrays corresponding to each recordset's field: `benefit`, `deduc_per_fam`, `pay_date`, and `reimbursement`.
- Lines 9 and 13 define the opening and closing tags describing an array named `benefit` corresponding to the field `benefit` in the recordset `FULL`.
- Lines 10-12 define the tags describing elements of array benefit including their name (row), type (string), and value.
- Lines 14-28 define the tags describing similar arrays and their elements corresponding to the other fields of the recordset `FULL`.
- Lines 30 and 45 define the opening and closing tags for the group of data representing the recordset named `BENEFITS`. In the following lines (31-44) the recordset `BENEFITS` is presented as a group of one-dimensional arrays corresponding to each field of a recordset.
- Line 47 represents similar blocks of data corresponding to the other job IDs.

Interactive Mode Processing – Example XML File

This section contains an example of an XML file used for interactive mode processing. The data here are equivalent to the data in the batch mode example.

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<interactive>
   <insured type="string">Full Time Employees</insured>
   <eff_date type="date">12/12/2012</eff_date>
   <pol_num type="double">66601</pol_num>
   <max_ben type="double">25000</max_ben>
   <FULL>
     <benefit type="array">
      <row type="string">1</row>
      <row type="string">2</row>
      <row type="string">3</row>
     </benefit>
     <deduc_per_fam type="array">
      <row type="string">25.00</row>
      <row type="string">50.00</row>
      <row type="string">100.00</row>
     </deduc_per_fam>
     <pay_date type="array">
      <row type="date">01/01/1999</row>
      <row type="date">03/31/1999</row>
      <row type="date">06/30/1999</row>
     </pay_date>
     <reimbursement type="array">
      <row type="double">80</row>
      <row type="double">100</row>
      <row type="double">120</row>
     </reimbursement>
   </FULL>
   <BENEFITS>
     <CODE type="array">
      <row type="double">1</row>
      <row type="double">2</row>
      <row type="double">3</row>
      <row type="double">4</row>
      <row type="double">5</row>
     </CODE>
     <RIDER type="array">
      <row type="string">Formulary Drug Plan</row>
      <row type="string">Hospital: Ward to Semi-Private</row>
      <row type="string">Supplemental Health Care</row>
      <row type="string">Vision Care</row>
      <row type="string">Intensive Care Rider</row>
     </RIDER>
   </BENEFITS>
 </interactive>
```

XPath InfoSource

XPath InfoSource dynamically translates IStream Query statements into XPath functions by mapping an IStream Query statement to an XPath string and executing the XPath string through the Microsoft XML Parser. Each XPath InfoSource is configured with a mapping file in XML that contains definitions to map all IStream statements to XPath strings.

In the same way as with a structured XML InfoSource, you initialize the XPath InfoSource with a `QUERY` statement as the first `QUERY` statement in an IStream Model, such as:

```
QUERY "File=c:\data\bookstores.xml", myxpath
```

Once the XPath InfoSource has been initialized, the subsequent `QUERY` statement will traverse the XML data tree to a destination node, as shown in this example:

XPath InfoSource Data Example

In this example, `Query "bookstore", myxpath` establishes the destination node to the `<bookstore>` node in this example:

```
<?xml version="1.0"?>
<bookstores>
 <bookstore num="100">
  <name>Best Chapters</name>
  <address>100 Main Street</address>
  <phone>999-123-8888</phone>
  <cafe>true</cafe>

  <book ibn="1001">
   <title lang="eng">Harry Potter</title>
   <author>J K Rowling</author>
   <year>2005</year>
   <price>29.99</price>
  </book>

  <book ibn="1002">
   <title lang="eng">Learning XML</title>
   <price>39.95</price>
  </book>
 </bookstore>
</bookstores>
```

At this point, all the elements and attributes under the destination node can be retrieved using references: see the reference definitions in this mapping file:

```
<?xml version="1.0" ?>
<mappings>
  <query name="bookstore" xpath="/bookstores/bookstore">
   <reference name="cafe" xpath="cafe" conversion="logic" />
   <reference name="store_id" xpath="@num" />
   <reference name="store_name" xpath="name"/>
   <reference name="book.author" xpath="book/author"/>
  </query>
</mappings>
```

Reference examples include:

- referring to the reference "store_id" returns "100" as a string
- referring to the reference "store_name" returns "Best Chapters"
- referring to the reference "book.author" returns "J K Rowling"

Preparing the XML File

In XPath InfoSources, the data for document generation are retrieved from an XML file containing data extracts from a database or another data source, such as a mainframe or legacy system. You need to prepare the XML file before generating documents.

This section includes guidelines for preparing the XML file for XPath InfoSource.

Structuring the XML File

Unlike the structured XML InfoSource, you do not need to structure the XML data file in any pre-defined way for XPath InfoSource. Any structure of the XML data is supported by the XPath InfoSource, as long as a mapping file is provided.

Note: Ensure that your XML file is well-formed. All the data provided in the XML file for document assembly must be formatted according to the syntax described in this chapter. The XPath InfoSource does not validate the completeness or correctness of the data in the XML file.

Mapping File Structure

The mapping file for XPath InfoSource is also an XML file. The general structure of the mapping file is:

```
<?xml version="1.0"?>
<mappings>
  <query name="query_name" xpath="xpath_expression" >
    <reference name="var_name" xpath="xpath_expression"
conversion="target_type" />
    …
  </query>
  …
  <lookup name="lookup_name" xpath="xpath_expression"
conversion="target_type" />
  …
  <checkup name="checkup_name" xpath="xpath_expression"
conversion="target_type" />
  …
  <reference … />
  …
</mappings>
```

Place all the definitions between the `<mappings>` tags.

The following tags are defined for XPath InfoSource mapping:

Query Name

```
<query name="query_name" xpath="xpath_expression">
```

This tag defines a query statement for an IStream Query. The `query_name` is the name referred to by an IStream Query statement. It is equivalent to the SQL `SELECT` statement referred by an IStream Query for an ODBC InfoSource. `query_name` is case-sensitive and must be unique within the mapping file. The `xpath_expression` defines an XPath expression for the query.

Please note:

- Use the absolute XPath expression starting from the XML document node for the xpath attribute (for example, `/bookstores/bookstore`) because of the precision of the absolute XPath expression.
- If a query is not a parameterized query, then the `name` attribute cannot contain any spaces. In a parameterized query, spaces are allowed in the `name` attribute.
- The nested `<query>` tags are no longer supported.

To pass a parameter to a query, you need to embed `{0}` in the `query_name` and the `xpath` expression as follows:

```
<query name="Query {0}" xpath="({0})" />
```

In your IStream Model document, you can code your IStream Query with a run-time string value, for example:

```
Define Query1 "/bookstores/bookstore"
Query "Query " + Query1, "XPathInfoSrc".
```

Currently, you can embed only one parameter in the XPath InfoSource `query_name`. For details, see *Mapping Example 3* on page 110.

If you use a parameterized query, the run-time string value passed to the parameter `{0}` cannot contain the reserved word `select`. This word cannot appear anywhere in the run time string, regardless of case (uppercase or lowercase). Because of this, your XML source data cannot use `select` as part of a tag name.

Reference Name

```
<reference name="var_name" xpath="xpath_expression"
conversion="target_type" />
```

This tag defines a variable reference for IStream. The `var_name` is the variable name to be referred to by IStream rules and expressions. It must be unique in the entire mapping file. The `xpath_expression` defines an XPath expression for this reference. If your XML data has a deep structure, you should use the relative XPath expression starting with its child tag (for example, `book/author`, without a single or double slash as the prefix - before book) to avoid getting the wrong data.

For the sample data in *XPath InfoSource Data Example* on page 103, the following references with different XPath expressions will get the same result but have different implications:

Reference Method 1

This reference searches for the immediate child (`book/author`) from the query destination node:

```
<reference name="book.author" xpath="book/author"/>
```

Reference Method 2

This reference searches for any sub-node (such as `book/author`) from the query destination node. It can be an immediate child node or a grandchild node:

```
<reference name="book.author" xpath="//book/author"/>
```

Reference Method 3

```
<reference name="book.author" xpath="/bookstore/book/
author"/>
```

This reference performs in the same way as the first reference. Because you are using an absolute XPath expression, the reference must start from the query destination node.

Specially, if a query definition (for example, its xpath attribute is `/bookstores/bookstore/book/price`) navigates its destination node to a leaf node in the XML data, then the only available references to be made are to the node itself (using the XPath `self` function (that is, `self::*`) and to any attributes of the node.

For example, the following equivalent reference definitions obtain the price from the previous query definition:

```
<reference name="book_price" xpath="self::*"
conversion="numeric" />
<reference name="book.price" xpath="/price"
conversion="numeric" />
```

The conversion attribute is an optional attribute (the default is `string`). Valid values are `string`, `date`, `logic` and `numeric`:

- `date` converts the reference value to IStream `DATE` type, similar to the IStream `CTOD` function. (See the `CTOD` function reference for details.)
- `logic` converts the reference value to an IStream `LOGICAL` type. The valid values for this type are `"true"` or `"false"` in any letter-case (uppercase or lowercase).
- `numeric` converts the reference value to IStream `NUMERIC` type, similar to the IStream `VAL` function. See the `VAL` function reference for details.

In addition, you should place all `<reference>` tags for a particular query between `<query>` and `</query>` in the query. These are called local references. All references defined in a particular `<query>` tag are global references, which are shared for all of queries. The reference names must be unique among the local references and the global references.

Lookup Name

```
<lookup name="lookup_name" xpath="xpath_expression"
conversion="target_type" />
```

This tag defines a lookup operation through the IStream lookup function for an XPath InfoSource. The lookup operation mimics the lookup operation for the ODBC InfoSource.

The name attribute always has the following exact format:

`Table=name0,Column0=name1,Column1=name2,Value={0}`

where `Table`, `Column0`, `Column1` and `Value= {0}` are reserved.

Note: The name attribute values are case-sensitive and cannot contain spaces.

The XPath attribute defines an XPath expression. The conversion attribute is an optional attribute which defines the data type for the lookup result. To define a meaningful lookup, you should embed `{0}` in the lookup xpath attribute.

Example

From the previous sample data, you can define a lookup using `ibn` as input and output the book price as follows:

```
<lookup name="Table=Book,Column0=ibn,Column1=price,Value={0}" xpath="/
bookstores/bookstore[@num=&quote;100&quote;]/
book[@ibn=&quote;{0}&quote;]/price" conversion="numeric" />
```

In the IStream model document, you can use the following `lookup` function to look up a value:

```
LOOKUP("1001", "Book", "ibn", "price", "XPath_InfoSrc")
```

Checkup Name

```
<checkup name="checkup_name" xpath="xpath_expression" />
```

This tag defines a checkup operation using the IStream `checkup` function for the XPath InfoSource. Like the `lookup` operation, the `checkup` operation mimics the checkup operation for the ODBC InfoSource.

The name attribute always has the following exact format:

`Table=name0,Column=name1,Value={0}`

where `Table`, `Column` and `Value={0}` are reserved.

Note: The name attribute values are case-sensitive and cannot contain any spaces.

The xpath attribute defines an XPath expression. To define a meaningful checkup, embed `{0}` in the checkup xpath attribute. For details, see the *LOOKUP and CHECKUP Functions* on page 118.

Complete Mapping Examples

The following complete mapping examples show how to prepare mapping files with XML sources for an XPath InfoSource.

Mapping Example 1 corresponds to XML Source Example 1. Mapping Example 2 to XML Source Example 2.

Mapping Example 1

In this example, three queries (`bookstore`, `bookstore2` and `bookstore3`) use three different referencing approaches (see the examples under *Reference Name* on page 106) and create the same results.

```
<?xml version="1.0" ?>
<mappings>
  <query name="bookstore" xpath="/bookstores/bookstore">
    <reference name="cafe" xpath="//cafe" conversion="logic" />
    <reference name="store_id" xpath="@num" />
    <reference name="store_name" xpath="//name"/>
    <reference name="book.author" xpath="//book/author"/>
  </query>

  <query name="bookstore2" xpath="/bookstores/bookstore">
    <reference name="cafe2" xpath="cafe" conversion="logic" />
    <reference name="store_id2" xpath="@num" />
    <reference name="store_name2" xpath="name"/>
    <reference name="book.author2" xpath="book/author"/>
  </query>

  <query name="bookstore3" xpath="/bookstores/bookstore">
    <reference name="cafe3" xpath="/bookstore/cafe" conversion="logic" />
    <reference name="store_id3" xpath="@num" />
    <reference name="store_name3" xpath="/bookstore/name"/>
    <reference name="book.author3" xpath="/bookstore/book/author"/>
  </query>
```

```
  <query name="bookprices" xpath="/bookstores/bookstore/book/price">
    <reference name="book_price" xpath="self::*" conversion="numeric" />
    <reference name="book.price" xpath="/price" conversion="numeric" />
  </query>

  <lookup name="Table=Book,Column0=ibn,Column1=price,Value={0}" xpath="/
bookstores/bookstore[@num=&quot;100&quot;]/book[@ibn=&quot;{0}&quot;]/
price" conversion="numeric" />
</mappings>
```

Mapping Example 2

This example shows how to use the XPath InfoSource to query the XML data prepared for the structured XML InfoSource.

In this example, all the XPath expressions are relative. The advantage of using a relative XPath expression is that this mapping does not change if the document tag name changes in the XML data. For example, if you change the `<interactive>` tag in the XML Source Example 2 to `<batch>`, the mapping is still valid.

```
<?xml version="1.0"?>
<mapping>
<query name="Select * from Policy" xpath="//Policy">
   <reference name="insured" xpath="//insured/row" conversion="string"/>
   <reference name="eff_date" xpath="//eff_date/row" conversion="date"/>
   <reference name="pol_num" xpath="//pol_num/row" conversion="string"/>
   <reference name="full_time" xpath="//full_time/row"
   conversion="logic"/>
   <reference name="max_ben" xpath="//max_ben/row" conversion="numeric"/>
   <reference name="week_indem" xpath="//week_indem/row"
    conversion="numeric"/>
   <reference name="beneficiary" xpath="//beneficiary/row"
    converison="string"/>
</query>

<query name="Select benefit, deduc_per_fam, reimbursement_1 from FULL"
xpath="//FULL"/>
<query name="Select * from FULL" xpath="//FULL"/>
<query name="Select deduc_per_fam from FULL" xpath="//FULL"/>
<reference name="benefit" xpath="//benefit/row" conversion="string"/>
<reference name="deduc_per_fam" xpath="//deduc_per_fam/row"
 conversion="string"/>
<reference name="reimbursement_1" xpath="//reimbursement_1/row"
 conversion="numeric"/>
<reference name="reimbursement_2" xpath="//reimbursement_2/row"
 conversion="numeric"/>
<reference name="pay_date_1" xpath="//pay_date_1" conversion="date"/>
<reference name="pay_date_2" xpath="//pay_date_2" conversion="date"/>
<lookup name="Table=BENEFITS,Column0=CODE,Column1=RIDER,Value={0}"
xpath="//BENEFITS/RIDER/row[number(//CODE/row[text()={0}])]"
conversion="string"/>
</mapping>
```

Mapping Example 3

This example shows how to use the XPath InfoSource with a parameterized string. This mapping works with data in *XML Source Example 1* on page 111

```
<?xml version="1.0" ?>
<mappings>
  <query name="Query {0}" xpath="({0})" />

    <reference name="cafe" xpath="child::cafe" conversion="logic" />
    <reference name="store_id" xpath="@num" />
    <reference name="store_name" xpath="child::name" />

    <reference name="title" xpath="child::title" />
    <reference name="price" xpath="child::price" />
    <reference name="all" xpath="self::*" />

</mappings>
```

The following IStream Model example shows how to pass the parameter to a parameterized query defined in the previous mapping example (Example 3).

```
*** Initialize Query
Query "", "XPathSample3"

DEFINE Query1 "/bookstores/bookstore"
DEFINE Query2 "//book"

Query "Query " + Query1, "XPathSample3"
TEXT
    Store ID = < store_id >
    Store Name = < store_name >
    Does store have a cafe = <cafe>
ENDTEXT
Next

Query "Query " + Query2, "XPathSample3"
TEXT
    Book Title = < title >
    Price = < price >
    All Book Content = <all>
ENDTEXT
Next

BREAK
NEXT
```

Complete XML Source Examples

XML Source Example 1

```
<?xml version="1.0"?>
<bookstores>
 <bookstore num="100">
  <name>Best Chapters</name>
  <address>100 Main Street</address>
  <phone>999-123-8888</phone>
  <cafe>false</cafe>

  <book ibn="1001">
   <title lang="eng">Harry Potter</title>
   <author>J K Rowling</author>
   <year>2005</year>
   <price>29.99</price>
  </book>

  <book ibn="1002">
   <title lang="eng">Learning XML</title>
   <price>39.95</price>
  </book>
 </bookstore>
</bookstores>
```

XML Source Example 2

```
<?xml version="1.0"?>
<interactive>
<Policy>
   <insured type="array">
      <row type="string">Full Time Employees</row>
   </insured>
   <eff_date type="array">
      <row type="date">12/12/12</row>
   </eff_date>
   <pol_num type="array">
      <row type="double">11111</row>
   </pol_num>
   <full_time type="array">
      <row type="boolean">True</row>
   </full_time>
   <max_ben type="array">
      <row type="double">25000</row>
   </max_ben>
   <week_indem type="array">
      <row type="double">60</row>
   </week_indem>
   <beneficiary type="array">
```

```
      <row type="string"></row>
   </beneficiary>
</Policy>
<FULL>

<benefit type="array">
   <row type="string">1</row>
   <row type="string">2</row>
   <row type="string">3</row>
   <row type="string">4</row>
   <row type="string">5</row>
</benefit>

<deduc_per_fam type="array">
   <row type="string">25.00</row>
   <row type="string"></row>
   <row type="string"></row>
   <row type="string">hello</row>
   <row type="string"></row>
</deduc_per_fam>

<pay_date_1 type="array">
   <row type="date">1/3/97</row>
   <row type="date">1/4/97</row>
   <row type="date">1/5/97</row>
   <row type="date">1/6/97</row>
   <row type="date">1/7/97</row>
</pay_date_1>

<pay_date_2 type="array">
   <row type="date">1/3/98</row>
   <row type="date">1/4/98</row>
   <row type="date">1/5/98</row>
   <row type="date">1/6/98</row>
   <row type="date">1/7/98</row>
</pay_date_2>

<reimbursement_1 type="array">
   <row type="double">80</row>
   <row type="double">100</row>
   <row type="double">100</row>
   <row type="double">125</row>
   <row type="double">100</row>
</reimbursement_1>

<reimbursement_2 type="array">
   <row type="double">0</row>
   <row type="double">0</row>
   <row type="double">0</row>
   <row type="double">150</row>
```

```
<row type="double">0</row>
</reimbursement_2>

</FULL>
<BENEFITS>
<CODE type="array">
   <row type="string">1</row>
   <row type="string">2</row>
   <row type="string">3</row>
   <row type="string">4</row>
   <row type="string">5</row>
   <row type="string">6</row>
   <row type="string">7</row>
   <row type="string">8</row>
</CODE>
<RIDER type="array">
   <row type="string">Formulary Drug Plan *</row>
   <row type="string">Hospital: Ward to Semi-Private</row>
   <row type="string">Supplemental Health Care</row>
   <row type="string">Vision Care</row>
   <row type="string">Out-of-Province Emergency</row>
   <row type="string">Coronary Or Intensive Care Rider</row>
   <row type="string">First Occurrence Rider</row>
   <row type="string">Cancer Disability - 1 Year -
   Individual</row>
</RIDER>
</BENEFITS>
</interactive>
```

About XPath

XPath is a language for finding information in an XML document. XPath (the XPath expression) is used to navigate through elements and attributes in an XML document. For more information, see the XML Path Language (XPath) Version 1.0 (W3C Recommendation - November 16, 1999) at www.w3.org/TR/xpath or see any other references about XPath.

After an XML file is parsed and loaded in the memory in a DOM structure, the parser accepts an XPath expression to retrieve parts of that structure. The result can be a single node or a set of nodes. A node can be used as context for running a new XPath expression. A resulting node can be a tag or the attribute of a tag.

The value of a resulting tag is usually the free text that follows the node tag, for example: `J K Rowling` in the tag `<author>J K Rowling</author>`.

- to refer to children `book` of a given context tag, the XPath expression is `book`
- to refer to grandchildren `author` of a given context tag, the XPath expression is `*/author`
- to refer to descendants `title` of a given context tag, the XPath expression is `//title`

- to refer to an attribute, use the @ character before its name; an attribute is the child of its tag node
- to refer to itself in a given context tag, the XPath expression is `self::*`

In an XPath statement, you can use predicates to refine the resulting set. A predicate can be an index or an expression. The expression can be as simple as comparing the value of a child attribute to a constant.

Example:
```
'*/book[2]', '(//book[@title="Learning XML"])/price)', '//
book[@ibn=”1001”]/author'
```

Adding XML InfoSources

The XML InfoSource is used by IStream Assembler during the document generation session to retrieve data from an XML file.

The XML InfoSource can support the use of non-customized XML with a mapping file and a standard XML source file, rather than the customized XML source files in previous versions. This is referred to as the XPath XML InfoSource.

Note: The XML InfoSource does not validate the completeness or correctness of the data structure in the XML file.

You add an XML InfoSource using InfoConnector. For details, see *Adding an IStreamXML InfoSource* in the *InfoConnector Online Help*.

The InfoSource can be used in two modes: default and canonical. To use the canonical, click the [...] button to browse for the XML Mapping file (for more information, see *Preparing the XML File* on page 96).

Coding Model Documents for XML InfoSources

You should code model documents somewhat differently to access data in an XML InfoSource.

Querying Data

This section describes how to query data from an XML InfoSource. It begins with the general syntax of the QUERY statement, and then presents more specific syntaxes for use in both batch and interactive processing.

When coding model documents for use with XML InfoSources, you usually only have to make one query to retrieve one block of data from the XML file related to the generation of a particular document. You can then query one or more documents' data directly from memory.

As a result, you do not need to use the following rules and functions that retrieve or access data directly from a database:

- INITIALIZE
- LOCATEITEM
- LOCATEFOUND
- LOCATENEXT
- FORCENEXT
- FIELD
- SELECT
- SELECT ALL

Important: When coding Model Documents for XML InfoSources, ensure that the same XML InfoSource is not referencing two different XML files during the same generation period.

General Syntax

Use the following general syntax for all your QUERY statements:

```
QUERY StringExpression[,InfoSourceName]
...
NEXT
```

In this statement:

- **StringExpression** is a mandatory parameter that can be an expression evaluating to a String type. It can be:
- a string **constant** (for example: `"JOBID=123"`),
- a name of a **String** variable (for example: id), or
- an **expression** (for example: `"FILE="+XMLFile`, where `XMLFile` is the name of a variable.)

- **InfoSourceName** can be either the name of the XML InfoSource, or a string expression evaluating to the name of such InfoSource. It is an optional parameter, provided it was previously specified in the USEIS rule.

Syntax Limitations

There is a limited subset of the SQL syntax supported in the SELECT statements used with the QUERY rule when coding model documents to be used in XML InfoSources. The QUERY rule only supports a very simple WHERE clause in the SELECT statement. As well, the following SQL syntax is not supported:

- Join queries to multiple record sets in the XML file
- Embedded SELECT statements
- Use of aliases for the field names

Note: QUERY rules in the model document may not be nested more than 18 levels deep

Querying All Data In The XML File

In either batch or interactive processing, you need to make a query to retrieve data from the XML file.

Note: Data remain defined only within the model code embraced by a QUERY/NEXT loop; therefore, reference it within the loop, not after it.

For Batch Processing

For batch processing, you can either query an XML InfoSource that is pre configured to point to an XML file, or you can define the XML filename in the statement itself.

Use the following syntax when the XML InfoSource points the XML file:

```
QUERY id[, XMLInfoSource]
or
QUERY "JOBID=number"[,XMLInfoSource]
...
BREAK
NEXT
```

In these statements:

- id is the parameter of the QUERY rule. Define parameter id as a key data variable.
- The actual variable name does not have to be defined exactly as id; it can be any other valid name). However, the value of this exact variable must be specified in the parameters of the Generate request.
- The matching NEXT rule must be preceded by the BREAK rule. Omitting the BREAK rule may cause errors in generation.

Use the following syntax when you want to define the XML filename in the model document code itself, rather than having the XML InfoSource point to the XML file:

```
QUERY "FILE=filename", XMLInfoSource
. . .
BREAK
NEXT
```

For Interactive Processing

For interactive processing, you can specify the path to the XML file and the XML InfoSource:

```
QUERY "FILE=filename", XMLInfoSource
...
BREAK
NEXT
```

In these statements:

- IStream Assembler creates an instance of XML InfoSource named **XMLInfoSource** and passes it the name of the XML file as its configuration parameter.
- FILE is the reserved word.
- XMLInfoSource uses filename to set the full path to the XML file specified (and overrides any pre configured path).

If you want the filename to be passed to IStream Assembler using a key data variable of a String type (for example, named **XMLFile**), use the following syntax:

```
QUERY "FILE="+XMLFile, XMLInfoSource
```

To use the path preconfigured in the XML InfoSource, use the following syntax:

```
QUERY "", XMLInfoSource
```

LOOKUP and CHECKUP Functions

XML InfoSources support the CHECKUP and LOOKUP functions applied to an XML InfoSource. These functions behave the same as when they are applied to an ODBC Database InfoSource.

However, when they are applied to an XML InfoSource, they must be coded using full syntax.

In the **CHECKUP** function, the **FieldName** parameter must be included:

```
CHECKUP (Value, RecordsetName, FieldName, XMLInfoSourceName)
```

In the **LOOKUP** function, the Field1 and Field2 parameters must be included:

```
LOOKUP (Value, RecordsetName, Field1, Field2, XMLInfoSourceName)
```

Note: In both cases, you can omit the **XMLInfoSourceName** parameter; IStream defaults to the predefined ODBC Database InfoSource “System”.

Example Model Document Code

The example model document code uses the coding guidelines outlined earlier in this section. It uses the InfoSource “XMLData”, pointing to the XML file.

```
QUERY id, "XMLData"
    TEXT
Policy number: <pol_num>
Effective Date: <DMY(eff_date)>
Insured category: <insured>
Maximal benefit: <DOL_AMT(max_ben)>
As an employee of XYZ Inc., the following Benefits are available
I.  Extended Health Care
```

Benefits	Deductible Per Family ($)	Reimbursement

```
    ENDTEXT+
    QUERY "Select * from FULL"
      TEXT
```

<ALLTRIM(LOOKUP(benefit, "BENEFITS","CODE","RIDER","XMLData"))>	<deduc_per_fam>	<MDY(pay_date)> <DOL_AMT (reimbursement)>

```
      ENDTEXT
    NEXT
    BREAK
    NEXT
```

Author Wizards and XML

This section describes how the IStream Author Wizards were updated to be able to read XML InfoSources/queries.

XML ISC Query Rule

If an XML ISC query rule does not follow the SQL `SELECT` statement syntax and is therefore not formed from individual `SELECT` statement elements (such as `FROM` and `WHERE`), then `SELECT` will be a value of the `NAME` attribute of the `query` element found in the XML ISC mapping file, as follows:

```
<query name="Select * from policy" xpath="."/>
```

This name has a free format and is used only to provide a mapping to the corresponding xpath statement (which does follow the xpath syntax). The following query is therefore valid:

```
<query name="PolicyQuery" xpath="."/>
```

This type of query rule implementation makes the SQL approach using separate fields for `SELECT`, `FROM` and `WHERE` elements unnecessary. Instead, a single field is associated with the list of all available fields through the mapping file query elements.

The XML query rule also requires an extra query statement (with matching `BREAK/NEXT` pairs) that is used to define a scope in the document where XML related variables are available.

XML Variables

In IStream Author documents, the scope of a specific XML ISC is defined by `QUERY-BREAK/NEXT` pair of statements. In this scope, all reference elements from the mapping file which are not query element child elements appear in the document as variables:

```
<reference name="PolicyPremium" xpath="//Transaction/
TotalPayableAmount"/>
```

These variables (for example, `POLICYPREMIUM`) are considered global. In addition to these elements, reference elements can reside under a query element and are considered to be local elements, and are available only in the scope of the specific query:

```
<query name="contextPolicyLocation" xpath="//Policy/
PolicyLocations/Location">

<reference name="LocationId" xpath="@Id"/>

</query>
```

For example, `LOCATIONID` variable is available only when `CONTEXTPOLICYLOCATION` is specified in the query rule.

The wizard screens for the XML ISC were adjusted to reflect these scope details.

Lookup and Checkup functions

Because XML ISC was introduced later than other main ISCs, it was designed to mimic similar existing ISCs (ODBC). It affected how ODBC-specific functions (such as LOOKUP and CHECKUP) were implemented in XML ISC. When the compiler sees the following function call in the Author document:

```
CHECKUP("5000111111", "account","number","XmlIscXXX")
```

it actually translates it (at generation runtime) into

```
Table=account,Column=number,Value={0}
```

string and looks for a checkup element in the XML ISC mapping file with the same name:

```
<checkup name="Table=account,Column=number,Value={0}"
   xpath="//account[number=&quot;{0}&quot;]"/>
```

Once the element is found, its xpath attribute value is used to perform actual operation using CHECKUP function first argument as a parameter:

```
//account[number="5000111111"]
```

Based on the result of this (X)query, the CHECKUP function result is formed.

A similar strategy is used for the LOOKUP function.

Allowed Values

The CHECKUP and LOOKUP functions used in the Author document cannot include an arbitrary combination of table and field values but can only include values that are predefined in the corresponding mapping file in the CHECKUP and LOOKUP elements.

The function wizards can contain bound fields. Whenever the CHECKUP and LOOKUP function parameters are requested, they are not filled individually (as with other ISCs and functions) but as a *bundle* of values. The user is presented with a choice of available CHECKUP and LOOKUP elements from the XML mapping file. A single element selection causes multiple function parameters to be populated at the same time.

Chapter 7

IStream Promoter

IStream Promoter is a tool that helps automate the promotion of model documents and their related sections, and Microsoft Word templates, from one environment (such as development, testing or production) into another.

IStream Promoter also allows you to convert Calligo Enterprise 5.4 model documents to the new IStream Document Manager format.

Important: After promoting documents, they will no longer be compatible with Calligo 5.4.

(For information about IStream Promoter, see *Step 4: Plan for IStream Promoter* on page 26 of the *IStream Document Manager Installation Guide*.)

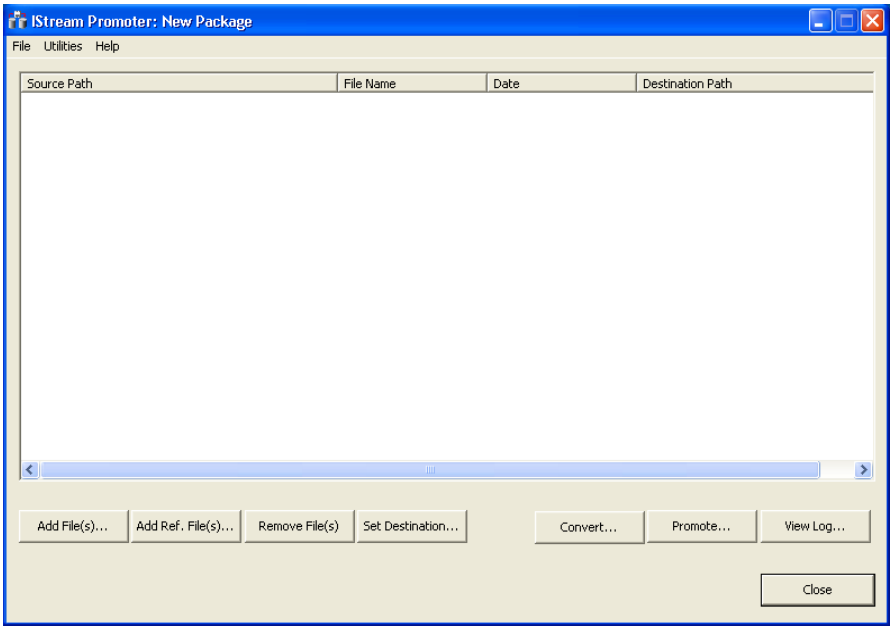
This chapter describes:

- *Opening IStream Promoter* on page 124
- *Creating a New File List* on page 125
- *Adding Files to the File List* on page 127
- *Adding Reference Files to the File List* on page 129
- *Removing Files from the File List* on page 131
- *Entering the File Destination Path* on page 132
- *Promoting or Converting the Files* on page 133
- *Viewing the IStream Promoter Log File* on page 134
- *Configuring IStream Promoter* on page 135

Opening IStream Promoter

- Click **Start > Programs > IStream > IStream Promoter**.

  IStream Promoter opens and displays the **New Package** window:



In this window, the source files are listed in a table that contains the following columns:

- **Source Path** – the folder containing the source files
- **File Name** – the name of the source files
- **Date** – the date that the files were last changed
- **Destination Path** – the destination folder path

You can now create a new file list.

Creating a New File List

In this step, you create a file list that will contain the files you want to promote. For each file to be promoted, you need to specify a source file and a destination location.

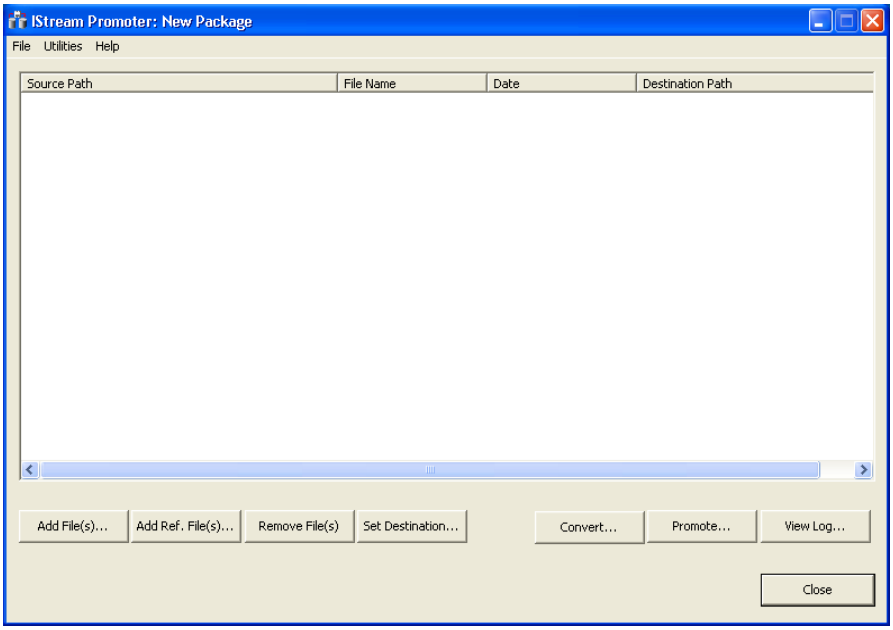
Note: The last file list that you were working on will automatically be loaded the next time you open IStream Promoter.

Method: Create and manage a new file list

---

1. In IStream Promoter, click **File > New**.

   The **New Package** window opens.



2. In this window, you can perform the following tasks:

   - **To add files to the list:** click **Add Files...**This step is covered in detail in *Adding Files to the File List* on page 127. Note that you can only promote IStream documents (.CMS, .CDS and .CLG files) and their associated templates (.DOT files).

To add reference files to the list: Select the files you want to add, then click **Add Ref Files...**(This step is covered in detail in *Adding Reference Files to the File List* on page 129.) Reference files are .CMS and .CDS sections that are included from the selected file(s), or .DOT files referenced by the selected file(s).

To remove files from this list: Select the files you want to remove, then click **Remove Files**. (This step is covered in detail in *Removing Files from the File List* on page 131.)

- **To change the destination path of a file:** Select the files that you want to change the destination of, then click **Set Destination**. (This step is covered in detail in *Entering the File Destination Path* on page 132.)
- **To sort the files in a different order:** Click the heading of the column on which you want to sort. Click the heading again to toggle the sort order between ascending and descending order.
- **To promote files and view the IStream Promoter log:** see *Promoting or Converting the Files* on page 133 and *Viewing the IStream Promoter Log File* on page 134.

3. When you are done making your changes, click **File > Save** to save the file list. To save the file list using a different name, click **File > Save As**, enter a new file name, then click **Save**.

Method: Open an existing file list

1. Click **File** > **Open.**

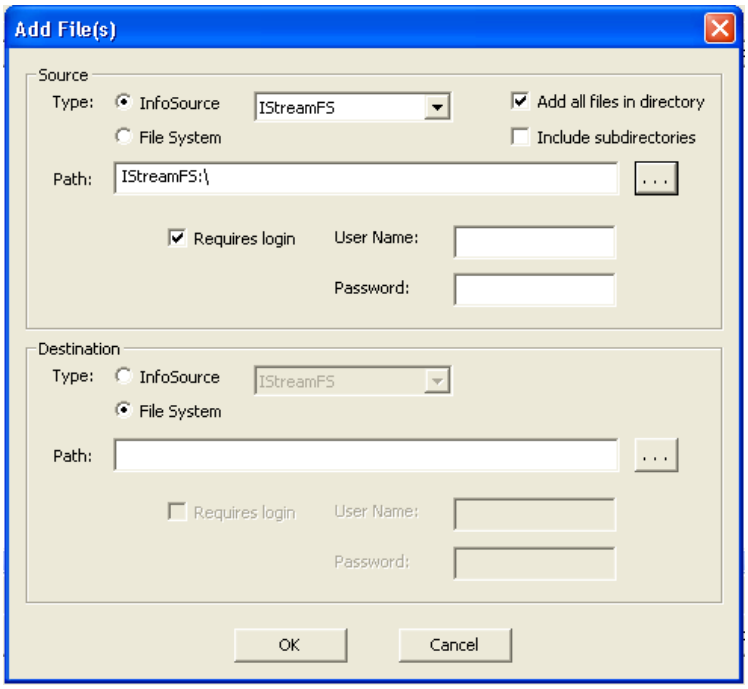
2. Browse to the file list that you want to open, then select it. Alternatively, select a file list from the list that appears at the bottom of the **File** menu.

Adding Files to the File List

In this step, you add files to the list of files that you want to promote.

Important: You can only promote IStream documents (.CMS, .CDS and .CLG files) and their associated templates (.DOT files).

1. After creating a new file list or opening an existing one, click **Add Files...**
2. The **Add Files** dialog opens.



3. In the **Source** section, select the repository **Type** of the source files: **InfoSource** or **File System**:
   - If you selected **InfoSource**, then select an **InfoSource** name from the drop-down list.

     If the **InfoSource** requires you to login, select the **Require Login** check box, then enter the required **User Name** and **Password**.
   - If you selected **File System**, continue to the next step.
4. If you are adding only one file, leave the **Add all files in directory** check box cleared, and continue to the next step.

   Alternatively, if you will be adding *all* the IStream model documents and Microsoft Word documents from a specific directory, select the **Add all files in directory** check box. If you also want to include the files within the subdirectories of this directory, then also select the **Include subdirectories** check box.
5. To enter a file or **Path**, click the browse button **[...]**.

   If you selected **Add all files in directory**, the **Select Folder** dialog opens.

If you did not check **Add all files in directory**, the **Select File** dialog opens.

6. Select the source file or folder, then click **OK**.
7. In the **Destination** section, select the repository type of the destination files: **InfoSource** or **File System**.
   - If you selected **InfoSource**, then select an **InfoSource** from the drop-down list.

     If the **InfoSource** requires you to login, select the **Require Login** check box, then enter the required **User Name** and **Password**.
   - If you selected **File System**, continue to the next step.
8. To enter a **Path**, click the browse button **[...]**.

   The **Select Folder** dialog opens.
9. Select the destination folder, then click **OK** to add the selected files to the file list.
10. After clicking **OK**, a message displays indicating that the files are being added.

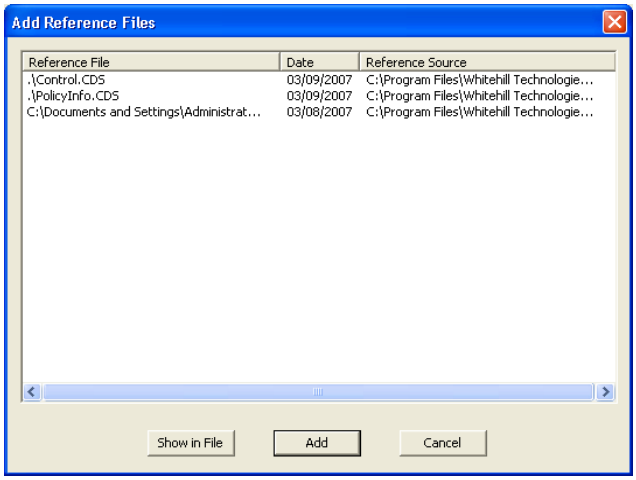
    If you try to add a file that is already in the current file list, you can choose to replace the existing file.
11. After adding the files, they will appear in the file list.

Adding Reference Files to the File List

Complete this step if you need to promote reference files. Reference files are .CMS and .CDS sections that are included from the selected file(s), or .DOT files referenced by the selected file(s).

1. Open a file list.
2. On the main window, select one or more .CMS or .CDS files.
3. Click **Add Ref. Files**.

   If any of the selected files contains references to other files, the **Add Reference Files** dialog opens.



   This dialog shows a list of referenced files in a table containing the following columns:

   - **Reference File** – the full file name and path
   - **Type** – the type of reference file: **IStream Section** or **Word Template**
   - **Date** – the date that the reference file was last modified
   - **Reference Source** – the full name of the file to which the **Reference File** belongs

4. To sort the list, click any column heading. To reverse the sort order, click the heading twice.

   To open the file list in Notepad, click **Show in File**. In Notepad, you can print or save the list.

5. Select the files that you want to add to the file list, then click **Add**.

Note: If you try to add a file that is already on the file list, you will be prompted to replace the existing file, skip the file or cancel the entire operation.

6. After you have finished adding the files, they will appear in the file list.

Note: If a reference file has a *relative* path, then after you add it to the file list, its path will be resolved based on the path of its parent file, and it will inherit its destination path from its parent file.

Removing Files from the File List

Complete this step if you need to remove files from the file list.

1. Select the files to remove from the file list.
2. Click **Remove Files**. A confirmation message box appears.
3. Click **OK** to remove the selected files from the file list.

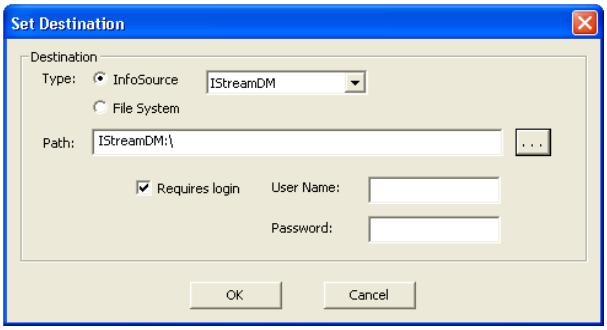
Entering the File Destination Path

Complete this step if you need to add or change the destination path.

Note: You do not need to set the destination path if you are converting files. If you leave this path blank, IStream Promoter will overwrite any existing CMS and CDS files and will not create copies in the destination folder.

1. After entering files for promotion, select the files which you want to set or change the destination path of.
2. Click **Set Destination**.

   The **Set Destination** dialog opens.



3. Select the repository type: **InfoSource** or **File System**.

   If you select **InfoSource:**

   - Select the **InfoSource** from drop-down list.
   - If the **InfoSource** requires a login, select the **Requires login** checkbox, then enter a **User Name** and **Password**.
4. To select the destination folder, click the browse button **[...]**.
5. Select a folder then click **OK**. The specified folder appears in the **Path**.
6. Click **OK** to save your changes.

Promoting or Converting the Files

After you have entered all the files to promote into the file list and configured the destination path, you can promote or convert the files.

Converting Files

The conversion feature allows you to convert Calligo Enterprise 5.4 model documents to the new IStream Document Manager format.

Note: You do not need to set the destination path if you are converting files. If you leave this path blank, IStream Promoter will overwrite any existing CMS and CDS files and will not create copies in the destination folder.

Method: Promote or convert files

---

1. Click **Promote** to promote the files, or click **Convert** to convert the model documents you have selected.

   IStream Promoter validates the files. A message box indicates the progress of the validation process.

   During validation, the paths are checked, as well as whether the source files exist.

   If the validation fails, the process stops. A message box indicates the failure. You can view the validation results in the log file: see *Viewing the IStream Promoter Log File* on page 134.
2. If the validation is successful, the **Promoter** dialog opens.
3. Select whether you want to keep the .CDS file extensions when promoting documents from a file system (FS) to a DMS.
4. To start the file promotion or conversion, click **Start**.

   A message box indicates the progress. To stop the process, click **Cancel**.
5. After the process has finished, a message box opens indicating if the process was successful or if it failed.

   If the process was successful, promoted files will be copied into destination locations as follows:

   - existing files on a **File System** will be overwritten by a new version
   - for existing files on a **DMS**, a new version of the file will be added

   If the process failed, you can view the log file: see *Viewing the IStream Promoter Log File* on page 134.

Viewing the IStream Promoter Log File

After promoting files, you can view the log file to see any validation or promotion errors.

- On the main window, click **View Log**.

  The IStream Promoter log file opens in Notepad.

This log file lists any files that have failed the validation or promotion processes. At the bottom of the log file is a summary of the promotion process, including how many files failed.

Note: To change the location of the log directory, see *Setting the Log File Directory* on page 135.

Configuring IStream Promoter

Use the **Utilities** menu to complete any of the following tasks:

- *Setting the Log File Directory* on page 135
- *Setting the Temporary Directory* on page 135
- *Purging the Temporary Files* on page 135

Setting the Log File Directory

You can enter the location where you want the IStream Promoter log file to be stored.

Method: Enter the location for the IStream Promoter log file

1. Click **Utilities > Set Log Directory...**
2. In the Browse for Folder dialog, double-click the folder you want to use to store the IStream Promoter log file.

Setting the Temporary Directory

You can enter the location where you want the IStream Promoter temporary files to be stored.

Method: Enter the location for the IStream Promoter temporary files

1. Click **Utilities > Set Temp Directory...**
2. In the **Browse for Folder** dialog, double-click the folder you want to use to store the IStream Promoter temporary files.

Purging the Temporary Files

Complete the following procedure to purge the IStream Promoter temporary files.

Method: Purge the IStream Promoter temporary files

- Click **Utilities > Purge Temp Files.**

  All IStream Promoter temporary files are deleted.

Chapter 8

System Tables

This chapter describes:

- *The System Tables* on page 138
- *The FORMATS Table* on page 139
- *The LANGTBL (Language) Table* on page 143

The System Tables

IStream Document Manager uses the following FORMATS and LANGTBL system tables at the generation time.

These tables can be modified by authorized users to meet specific business needs.

The tables are installed both on the Assembly Server host computer and on each Author Workstation client computer in the following Access 97 directory:

```
RootDirectory\Components\Shared
```

where RootDirectory is the directory where the corresponding IStream components (Assembly Server or Author Workstation) were installed.

These tables are accessed through a predefined InfoSource called System. It is an ODBC database type InfoSource. This InfoSource has a configuration parameter called Data Source set to “system”, which is the name of the ODBC System DSN. This ODBC Data Source uses Microsoft Access ODBC driver and points to the calsystem.mdb file mentioned above.

If for some reason (for example, maintenance), the format of system tables shipped with IStream Document Manager is not appropriate, they can be converted to a different database format or included in some other database as long as the following conditions are maintained:

- Table names have to be preserved equal to the original names.
- ODBC DSN has to be created pointing to the database containing converted (imported) IStream Document Manager system tables.
- System InfoSource has to be reconfigured to point to that new DSN.

A description of the IStream Document Manager system tables follows. To modify these tables, users must use an appropriate tool such as Microsoft Access, which allows one to load and edit the tables.

The FORMATS Table

This table contains specifications for different (including custom or user defined) formats for dates and currencies. IStream Document Manager language functions uses the formats to format date and currency data that are inserted into the generated document.

For example, when IStream Document Manager is originally installed, the default dollar value format is $1,000.00 (with a space after the dollar sign). If you want your documents to print dollar values as $1,000.00 (without the space), you can do so by editing the FORMATS table.

This table has the following design (including the description of each field).

Fields	Data Type	Field Size	Description
FORMATLANG	Text	12	Name of record
EXTRA	Text	10	Explanation of record
JANLONG FEBLONG MARLONG APRLONG MAYLONG JUNLONG JULONG AUGLONG SEPLONG OCTLONG NOVLONG DECLONG	Text	10	How each month will appear in long format

Fields (Continued)	Data Type	Field Size	Description
JANSHORT FEBSHORT MARSHORT APRSHORT MAYSHORT JUNSHORT JULSHORT AUGSHORT SEPSHORT OCTSHORT NOVSHORT DECSHORT	Text	10	How each month will appear in short format
SUNLONG MONLONG TUELONG WEDLONG THULONG FRILONG SATLONG	Text	10	How each day will appear in long format
SUNSHORT MONSHORT TUESHORT WEDSHORT THUSHORT FRISHORT SATSHORT	Text	10	How each day will appear in short format
DMYDAY	Text	1	How the day will appear when the DMY function is used
DMYMONTH	Text	1	How the month will appear when the DMY function is used
DMYYEAR	Text	1	How the year will appear when the DMY function is used
DMYFORM	Text	10	Format the date will appear when the DMY function is used
DMYSEP	Text	1	The separator used when the DMY function is used

Fields (Continued)	Data Type	Field Size	Description
MDYDAY	Text	1	How the day will appear when the MDY function is used
MDYMONTH	Text	1	How the month will appear when the MDY function is used
MDYYEAR	Text	1	How the year will appear when the MDY function is used
MDYFORM	Text	10	Format the date will appear when the MDY function is used
MDYSEP	Text	1	The separator used when the MDY function is used
YMDDAY	Text	1	How the day will appear when the YMD function is used
YMDMONTH	Text	1	How the month will appear when the YMD function is used
YMDYEAR	Text	1	How the year will appear when the YMD function is used
YMDFORM	Text	10	Format the date will appear when the YMD function is used
YMDSEP	Text	1	The separator used when the YMD function is used
CURFORM	Text	20	The format that currency will appear in when the DOL_AMT or NUMBER function is used
CURSYMBOL	Text	5	The currency symbol used when the DOL_AMT or NUMBER function is used
CURSEP	Text	3	The separator used when either the DOL_AMT or NUMBER function is used
CURDEC	Text	3	The decimal format when either the DOL_AMT or NUMBER function is used
CUR2FORM	Text	20	The format that currency will appear in when the DOL_AMT2 or NUMBER2 function is used
CUR2SYMBOL	Text	5	The currency symbol used when the DOL_AMT2 or NUMBER2 function is used
CUR2SEP	Text	3	The separator used when either the DOL_AMT2 or NUMBER2 function is used
CUR2DEC	Text	3	The decimal format when either the DOL_AMT2 or NUMBER2 function is used.

If you need a format that does not appear in the FORMATS table, create a custom format.

For example, if you want different date formats in your English contract than in your English booklet, you would need to add a new record which supports the second date format. For this new record, you would set the value for field `Formatlang` to *`English2`*.

To use this new format in a section, it is necessary to use following statement:

```
CALL SETFORMAT("English2")
```

After changes to FORMATS table are done and tested on the Author workstation, they have to be replicated (or copied) in the same table on the Assembly Server's host computer to make them available for remote generation.

The LANGTBL (Language) Table

This table contains specifications of different language names and their corresponding codes which can be used by IStream Document Manager rules and functions during the generation.

Table LANGTBL has the following design (including the description of each field).

Field	Data Type	Field Size	Description
LANGCODE	Text	1	Code of the language
LANGNAME	Text	15	Name of the language

LANGTBL table shipped with IStream Document Manager has the following records.

LANGCODE	LANGNAME
e	English
f	French
g	German
i	Italian
s	Spanish

For example, to add another version of English - American - to this table, you will have first to change the size of field LANGCODE from one to 2. After that add a new record with LANGCODE = E2 and LANGNAME = American.

To use this new language in a section, you would write a code a statement like the following one:

```
CALL SETLANGUAGE("American")
TEXTE2
This text will appear only for Language = "American"
ENDTEXT
```

INDEX

A

adding
- items, 44

administering
- and permissions, 34
- DMS, 44

Assembler, 15
- administration, 49
- ID, 38
- server, 29
  - making available for generation, 49
  - running as an application or service, 51

attributes, 45
audit trail, 63
- purging, 64
- querying, 64

Author, 38
- workstation, 28

automatic TOC update registry setting, 88

B

batches
- QUERY, 117

BREAK, 117
browser configuration, 25

C

cache folder, 60
- maintenance, 60

cached files, 61
caching documents, 59
Calligo
- servlet
  - configure, 74
  - generate, 82
- servlet.generate, 84

Calligoservlet.generate, invoking, 84
categories, 45
CHECKUP, 118
client components, 15
coding model documents
- changes, 116
- example, 119

convert files using Promoter, 133
creating
- users, 41

D

defined variables, 48
deleting
- groups, 41
- users, 43

DMS administration, 44
document
- caching, 59
  - limitations, 61
- generation, 115
- ID, 61
- promoting, 65

document conventions, 12

E

editing
- groups, 41
- users, 43

embedding fonts, PDF, 89
Enterprise workspace structure, 44
errors, 39
example
- model document code, 119

F

FIELD, 116
file destination, IStream Promoter
- IStream Promoter
  - file destination, entering, 132

file extensions, 91
file lists, creating, 125
file names, 91
files, adding to IStream Promoter, 127
files, removing from IStream Promoter file list, 131
font embedding, 89
font embedding options, 89
FORCENEXT, 116
formats table, 139

G

generation
- mode, 50
- using Publisher, 82

groups, 40
- creating, 40
- deleting, 41
- editing, 41

groups, creating, 40

I

IDs used in processing, 38

InfoConnector, overview, 15
InfoSourceName, 117
InfoSources
  administration, 48
  configuration, 32
  configuring system, 30
  creation and set up, 30
  installing, 30
INITIALIZE, 116
interactive use
  QUERY, 118
interactiveservlet, 79
interactiveservlet.configure, 74
IStream Document Manager
  deployment, 28
  items, 45
  items, adding, 44
  parameters, 46
IStream Promoter
  adding files to, 127
  adding reference files, 129
  configuring, 135
  creating new file lists, 125
  file lists, 127
  log file directory, setting, 135
  log file, viewing, 134
  opening, 124
  promoting files, 133
  removing files from, 131
  temporary directory, setting, 135
  temporary files, purging, 135
IStream Publisher, 82
item properties, 44

L

language Table, 143
limitations on document caching, 61
listing users and groups, 40
LOCATEFOUND, 116
LOCATEITEM, 116
LOCATENEXT, 116
log
  files, 56
  message types, 55
  viewer, 54
  viewer, optimizing, 54
log file directory, IStream Promoter, 135
log file, IStream Promoter, 134
logging
  function, 57
  level, changing, 56
LOOKUP, 118

M

managing sessions, 34
margins, PDF, 88
message filters, 56
model, 18, 35
model documents, changes in coding, 116

N

new group, 40
new installation, planning, 32
NEXT, 117
NextURLSuccess example, 84
NonInteractiveServlet, 81
NonInteractiveServlet.configure, 76
notes ID, 38

O

opening
  log viewer, 54
optimizing for performance, 52
overview, 14

P

parameters, configuring, 46
PDF defaults
  margins, 88
  TOC update, 88
PDF, configuring font embedding options, 89
permissions, 34
planning
  new installation, 32
print feature, 91
promoting documents, 65
promoting files, 133
proxy user IDs, 37
purging audit trail, 64

Q

QUERY, 116
  for batch processing, 117
  for interactive processing, 118
  general syntax, 116
querying the audit trail, 64

R

reference files, adding to IStream Promoter, 129
requirements, system, 25
reserved words, 67
roles, 20
running Assembly server as an application or service, 51

S

security, 17

and permissions, 34
and permissions with proxy IDs, 38
information propagation, 36
model, 34
SELECT, 116
SELECT ALL, 116
servlet dependency, 72
session management, 34
setting up
generation mode, 50
StringExpression, 116
synchronization on cached file, 61
system
category, 46
model, 18
requirements, 25
tables, 138

T

tables, system, 138
temporary directory, IStream Promoter, 135
temporary files, purging, 135
TOC update, PDF, 88

U

unified model, 35
unified model for IDM, 35
users, 40
administration, 40
and groups, listing, 40
creating, 41
deleting and editing, 43
IDs, proxy, 37
profiles, 41

V

variables, defined, 48
version
identifying, 66
number, 61
version, identifying, 66
viewing log files, 56

W

web ID, 38
word
processors, 16
workflows, 14
workspace structure, 44
workstation
applications, 35
configuration, 28
workstations, configuring, 28

X

XML file
path, 118

Z

zip & download, 91