Actuate One™

One Design
One Server
One User Experience

Installing BIRT iHub for Windows

Information in this document is subject to change without notice. Examples provided are fictitious. No part of this document may be reproduced or transmitted in any form, or by any means, electronic or mechanical, for any purpose, in whole or in part, without the express written permission of Actuate Corporation.

© 1995 - 2013 by Actuate Corporation. All rights reserved. Printed in the United States of America.

Contains information proprietary to:
Actuate Corporation, 951 Mariners Island Boulevard, San Mateo, CA 94404

www.actuate.com

The software described in this manual is provided by Actuate Corporation under an Actuate License agreement. The software may be used only in accordance with the terms of the agreement. Actuate software products are protected by U.S. and International patents and patents pending. For a current list of patents, please see http://www.actuate.com/patents.

Actuate Corporation trademarks and registered trademarks include:
Actuate, ActuateOne, the Actuate logo, Archived Data Analytics, BIRT, BIRT 360, BIRT Analytics, The BIRT Company, BIRT Data Analyzer, BIRT iHub, BIRT Performance Analytics, Collaborative Reporting Architecture, e.Analysis, e.Report, e.Reporting, e.Spreadsheet, Encyclopedia, Interactive Viewing, OnPerformance, The people behind BIRT, Performancesoft, Performancesoft Track, Performancesoft Views, Report Encyclopedia, Reportlet, X2BIRT, and XML reports.

Actuate products may contain third-party products or technologies. Third-party trademarks or registered trademarks of their respective owners, companies, or organizations include:
Mark Adler and Jean-loup Gailly (www.zlib.net): zLib. Adobe Systems Incorporated: Flash Player. Amazon Web Services, Incorporated: Amazon Web Services SDK, licensed under the Apache Public License (APL). Apache Software Foundation (www.apache.org): Ant, Axis, Axis2, Batik, Batik SVG library, Commons Command Line Interface (CLI), Commons Codec, Crimson, Derby, Hive driver for Hadoop, Pluto, Portals, Shindig, Struts, Tomcat, Xalan, Xerces, Xerces2 Java Parser, and Xerces-C++ XML Parser. Castor (www.castor.org), ExoLab Project (www.exolab.org), and Intalio, Inc. (www.intalio.org): Castor. Day Management AG: Content Repository for Java. Eclipse Foundation, Inc. (www.eclipse.org): Babel, Data Tools Platform (DTP) ODA, Eclipse SDK, Graphics Editor Framework (GEF), Eclipse Modeling Framework (EMF), and Eclipse Web Tools Platform (WTP), licensed under the Eclipse Public License (EPL). Gargoyle Software Inc.: HtmlUnit, licensed under Apache License Version 2.0. GNU Project: GNU Regular Expression, licensed under the GNU Lesser General Public License (LGPLv3). HighSlide: HighCharts. Jason Hsueth and Kenton Varda (code.google.com): Protocole Buffer. IDAutomation.com, Inc.: IDAutomation. IDRsolutions Ltd.: JBIG2, licensed under the BSD license. InfoSoft Global (P) Ltd.: FusionCharts, FusionMaps, FusionWidgets, PowerCharts. Matt Inger (sourceforge.net): Ant-Contrib, licensed under Apache License Version 2.0. Matt Ingenthron, Eric D. Lambert, and Dustin Sallings (code.google.com): Spymemcached, licensed under the MIT OSI License. International Components for Unicode (ICU): ICU library. jQuery: jQuery, licensed under the MIT License. Yuri Kanivets (code.google.com): Android Wheel gadget, licensed under the Apache Public License (APL). LEAD Technologies, Inc.: LEADTOOLS. The Legion of the Bouncy Castle: Bouncy Castle Crypto APIs. Bruno Lowagie and Paulo Soares: iText, licensed under the Mozilla Public License (MPL). Microsoft Corporation (Microsoft Developer Network): CompoundDocument Library. Mozilla: Mozilla XML Parser, licensed under the Mozilla Public License (MPL). MySQL Americas, Inc.: MySQL Connector. Netscape Communications Corporation, Inc.: Rhino, licensed under the Netscape Public License (NPL). OOPS Consultancy: XMLTask, licensed under the Apache License, Version 2.0. Oracle Corporation: Berkeley DB, Java Advanced Imaging, JAXB, JDK, Jstl. PostgreSQL Global Development Group: pgAdmin, PostgreSQL, PostgreSQL JDBC driver. Progress Software Corporation: DataDirect Connect XE for JDBC Salesforce, DataDirect JDBC, DataDirect ODBC. Rogue Wave Software, Inc.: Rogue Wave Library SourcePro Core, tools.h++. Sam Stephenson (prototype.conio.net): prototype.js, licensed under the MIT license. Sencha Inc.: Ext JS, Sencha Touch. ThimbleWare, Inc.: JMemcached, licensed under the Apache Public License (APL). World Wide Web Consortium (W3C) (MIT, ERCIM, Keio): Flute, JTidy, Simple API for CSS. XFree86 Project, Inc.: (www.xfree86.org): xvfb. ZXing authors (code.google.com): ZXing, licensed under the Apache Public License (APL).

All other brand or product names are trademarks or registered trademarks of their respective owners, companies, or organizations.

Document No. 130131-2-430342 April 10, 2013

Contents

Part 2
Installing

Understanding ActuateOne

ActuateOne™ includes iHub of Actuate® Corporation's value-added products for the Eclipse BIRT open source project. ActuateOne institutes a paradigm shift in Business Intelligence technology from individualized tools to a suite of integrated capabilities within a single environment. ActuateOne is one designer, one server, one integrated environment providing a single unified user experience. A common architecture is precisely what today's information-rich global environment requires for development and deployment. This unified Actuate technology continues to enable information management and delivery while supporting advanced security, massive scalability, flexibility through programming, and reuse.

ActuateOne realizes our corporate vision of a single user experience by providing extended new analytics capabilities reaching a broader spectrum of users. The new dashboard functionality supports building gadgets to enhance the visual presentation of information. Export to Excel® and other formats integrates Actuate output with other tools on the end-user desktop. Actuate's cloud-ready server supports elastic clustering for dynamic provision of uninterrupted efficient service.

Information, live demos, and endorsements about this release are available from birt-exchange.com and actuate.com. The Actuate site also makes "The Forrester Wave™: Open Source Business Intelligence (BI), Q3 2010" report freely available. The report recognizes Actuate and its value-added offerings for BIRT as a leader in Open Source Business Intelligence.

About the Installing BIRT iHub for Windows documentation

The printed and online documentation includes the materials described in Table I-1. You can obtain HTML and PDF files from the BIRT Exchange or Actuate Customer Support site.

Documentation updates are created in response to customer requirements and are available at both sites.

Table I-1 BIRT iHub documentation

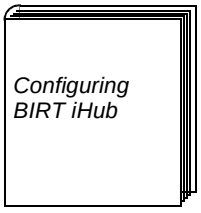
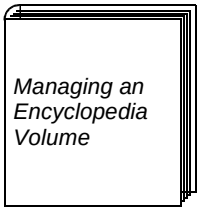
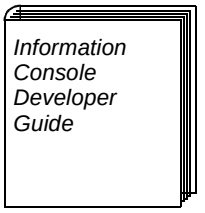
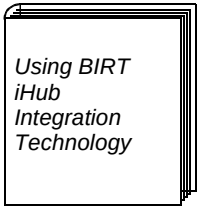
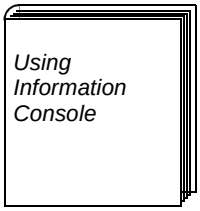
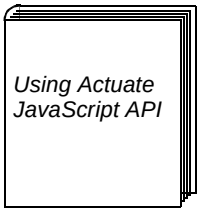
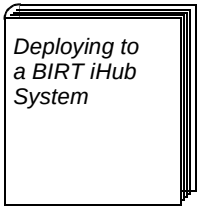
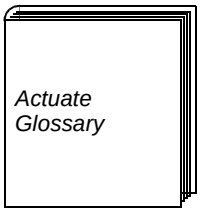
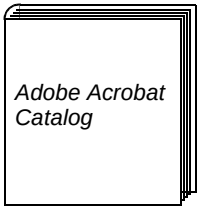
For information about this topic	See the following resource
Installing BIRT iHub for Linux	
Installing BIRT iHub for Windows	
Late-breaking information and documentation updates	Release notes and updated localization, HTML help, and PDF files posted on birt-exchange.com and Actuate Support
Configuring BIRT iHub Use Configuration Console to: ■ Add additional Encyclopedia volumes ■ Configure clusters of iHubs ■ Tune iHub services and processes ■ Configure e-mail notification ■ Review and update license options ■ Open ports for iHub use ■ Manage iHub printers and resources ■ Configure diagnostic logging	

Table I-1 BIRT iHub documentation (continued)

For information about this topic	See the following resource
Managing an Encyclopedia Volume Use Management Console to: ■ Set up user accounts ■ Set up channels and notification groups ■ Assign security roles ■ Manage files and folders ■ Schedule, run, and manage reports ■ Back up the Encyclopedia volume ■ Use Actuate Open Security	
Information Console Developer Guide ■ Overview of Information Console concepts and web applications ■ Using, customizing, and configuring the Deployment Kit ■ Using code components for JSPs, URL parameters, JavaScript files, Java servlets, Java Beans, and security facilities	
Using BIRT iHub Integration Technology ■ Overview of Actuate web services and SOAP messaging ■ Managing an Encyclopedia volume ■ Developing API applications using Java or .NET ■ Using Java Report Server Security Extension (RSSE) APIs ■ Using logging, performance monitoring, and archiving features ■ Customizing the Actuate software installation process ■ Actuate Information Delivery API operations and data types reference	

(continues)

Table I-1 BIRT iHub documentation (continued)

For information about this topic	See the following resource
Using Information Console ■ Overview of Information Console concepts and online reporting ■ Accessing and managing files and folders; running reports	
Using Actuate JavaScript API ■ Overview of programming with Actuate JavaScript ■ Creating custom pages using Actuate JavaScript ■ Reference for Actuate JavaScript classes and methods	
Deploying to a BIRT iHub System Describes how to deploy designs and information objects to iHub	
Actuate Glossary Definitions of product terminology	
Adobe Acrobat Catalog A utility that can search all the documents in the Actuate manuals directory	

Obtaining documentation

Actuate provides technical documentation in PDF, HTML, and print formats. You can download PDF or view HTML versions of the documentation from birt-exchange.com. If you purchase the product, you can also download documentation using ftp as instructed in the e-mail from Actuate Distribution. When you install the files using the Online Documentation and Localization Resource Files program, if you accept the default location, the program loads the PDF in the Actuate\Manuals directory.

Using PDF documentation

In each PDF version of a book, the table of contents and the index page numbers contain links to the corresponding topics in the text. In the table of contents, you access the link by positioning the pointer over the topic. In the index, you access the link by positioning the pointer over the page number.

The Program Files (x86)\Actuate\Manuals directory contains a file, master-index.pdx, which is an Adobe Acrobat Catalog utility that can search all the documents in the Actuate Manuals directory. This tool provides a convenient way to find information on a particular topic in Actuate documentation.

Obtaining late-breaking information and documentation updates

The release notes contain late-breaking news about Actuate products and features. The release notes are available on the Actuate Support site at the following URL:

`http://support.actuate.com/documentation/releasenotes`

Updates to documentation in PDF form are available at the following URL:

`http://support.actuate.com/documentation`

If you are a new user, you must first register on the site and log in to view the release notes. Birt-exchange.com and actuate.com also provide product update information.

About obtaining technical support

You can contact Customer Support by e-mail or telephone. For contact information, go to the following URL:

`http://www.actuate.com/services/support/contact-support.asp`

About supported and obsolete products

The Actuate Support Lifecycle Policy and Supported Products Matrix are available on the Actuate Support web site at the following URL:

http://support.actuate.com/documentation/spm

Typographical conventions

Table I-2 describes the typographical conventions in this document.

Table I-2 Typographical conventions

Item	Convention	Example
Code examples	Monospace	`Dim Text1 As String`
File names	Initial capital letter, except where file names are case-sensitive	Detail.roi
Key combination	A + sign between keys means to press both keys at the same time	Ctrl+Shift
Menu items	Capitalized, no bold	File
Submenu items	Separated from the main menu item with a small arrow	File➤New
User input or user response	Monospace	`M*16*`
User input in XML and Java code	Monospace italics	`chkjava.exe` *`cab_name`*`.cab`

Syntax conventions

Table I-3 describes the symbols used to present syntax.

Table I-3 Syntax conventions

Symbol	Description	Example
[ ]	Optional item	[Alias<alias name>]
	Array subscript	matrix[ ]

Table I-3 Syntax conventions

Symbol	Description	Example
{ }	Groups two or more mutually exclusive options or arguments when used with a pipe	{While | Until}
	Defines array contents	{0, 1, 2, 3}
	Delimiter of code block	public ACJDesigner( ){ }
|	Separates mutually exclusive options or arguments in a group	Exit {Do | For | Function | Sub}
	Java OR operator	int length | 4
< >	Argument you must supply	<expression to format>
	Delimiter in XML	<xsd:sequence>

About *Installing BIRT iHub for Windows*

Installing BIRT iHub for Windows includes the following chapters:

- *Introduction*. Provides an overview of this guide, BIRT iHub documentation, and the typographical conventions in this book.
- *Part 1. Architecture.* Describes BIRT iHub architecture.
- *Chapter 1. Understanding Actuate BIRT iHub architecture.* Describes BIRT iHub architecture, the iHub System process model, and system administration, including new utilities and third-party relational database management systems (RDBMS) used to store iHub system and Encyclopedia volume metadata.
- *Part 2. Installing.* Describes how to install BIRT iHub.
- *Chapter 2. Installing BIRT iHub.* Describes how to install BIRT iHub using the out-of-the-box (OOTB) PostgreSQL relational database management system (RDBMS) in a Windows environment.
- *Chapter 3. Installing BIRT iHub using an alternative database.* Describes how to install BIRT iHub using an alternative RDBMS, such as a pre-existing PostgreSQL or Oracle RDBMS, in a Windows environment.
- *Chapter 4. Migrating to BIRT iHub.* Describes how to upgrade BIRT iHub in a Windows environment.

- *Chapter 5. Installing a BIRT iHub cluster.* Describes how to install a BIRT iHub cluster node in a Windows environment.
- *Chapter 6. Installing BIRT iHub in a cloud.* Describes how to install BIRT iHub in a cloud environment using a ready-to-launch iHub image.
- *Chapter 7. Installing Information Console.* Describes how to install Actuate Information Console in Windows.
- *Chapter 8. Installing iHub Integration Technology and documentation.* Describes how to install Actuate iHub Integration Technology and Documentation in a Windows environment.
- *Part 3. Licensing.* Describes how to license BIRT iHub.
- *Chapter 9. Licensing BIRT iHub.* Describes licensing options, license key installation, and CPU-binding policies for BIRT iHub.
- *Part 4. Backing Up.* Describes how to back up a BIRT iHub.
- *Chapter 10. Backing up an Encyclopedia volume.* Describes how to back up and restore BIRT iHub Encyclopedia volume metadata and data.
- *Part 5. Utilities.* Describes how to use iServer and iHub utilities.
- *Chapter 11. Working with BIRT iHub utilities.* Describes how to perform administration tasks on system and Encyclopedia schemas using the iServer and iHub utilities.

Part One

Architecture

Chapter

1

Understanding Actuate BIRT iHub architecture

This chapter contains the following topics:

- Understanding BIRT iHub architecture
- Understanding the iHub System process model
- Administering iHub System

Understanding BIRT iHub architecture

Actuate BIRT iHub stores metadata containing information about the system and Encyclopedia volume configuration in a relational database management system (RDBMS). In an out-of-the-box (OOTB) installation, Actuate BIRT iHub uses a customized version of the open-source, third-party database, PostgreSQL. iHub also supports using other alternative, third-party database systems, such as DB2, Microsoft SQL Server, Oracle, or a pre-existing PostgreSQL instance.

iHub stores metadata in the following schemas:

- System
  Contains settings related to iHub configuration, such as servers, templates, volumes, and partitions.
- Encyclopedia volume
  Contains settings related to volume configuration, such as users, roles, groups, channels, folders, files, and other objects.

In iHub, Actuate provides the following installation options:

- Install a new iHub system with a PostgreSQL or other supported, alternative, third-party database.
- Upgrade an existing Actuate iServer installation from a previous release, such as Release 10 Service Pack 1, to Release 11 Service Pack 4, then migrate to iHub using a side-by-side installation operation.
- Migrate an existing Release 11 Service Pack 4 installation to iHub using a side-by-side installation operation.

Using a third-party RDBMS with an Encyclopedia volume

Actuate automatically installs the iHub system and Encyclopedia volume schemas in the OOTB PostgreSQL RDBMS installation. Installing the schemas in a pre-existing PostgreSQL RDBMS or alternative RDBMS, such as DB2, Oracle, or SQL Server, requires manually running a SQL script containing the appropriate Data Definition Language (DDL) statements. The installation sections of this book contain chapters that provide detailed, step-by-step descriptions on how to perform these operations.

Actuate BIRT iHub only supports installing the metadata for Encyclopedia volumes in the same schema, not separate schemas or separate databases. In Oracle, there is a one-to-one relationship between a database user and a schema. A schema is not a separate entity. An Actuate DB2, PostgreSQL, or SQL Server installation requires this one-to-one relationship between a database user and a schema for consistency.

In a pre-existing RDBMS installation, the database administrator first creates a schema owner and a database user by running a SQL DDL script. During iHub installation, the iHub system administrator provides the schema owner and database user credentials. The iHub installation program connects to the RDBMS, creates the necessary Encyclopedia volume database structures, then loads the metadata. The iHub system interacts with the third-party RDBMS using these database user credentials.

Only the metadata that specifies the Encyclopedia volume configuration are in the database. Designs, documents, information objects, and other iHub data objects are stored in the file system.

Customizing Encyclopedia volume databases

Actuate supports read-only operations on the system and Encyclopedia volume metadata in the tables of the OOTB or other third-party database. Actuate does not support the addition, deletion, or modification of these metadata tables.

Actuate does permit the creation of additional indexes on these tables. For example, a customer can create an index on the job completion notices table to expedite database processing.

Actuate does not recommend any customization of the system metadata schema. Any customization that the customer does on the Encyclopedia volume schema must be redone when migrating to, reinstalling, or upgrading iHub. Actuate iHub does not track the objects that a customer creates. Actuate reserves the right to change the structure of the schema in future releases.

Installing and configuring iHub System

The installation, configuration, and administration of an iHub System can include the following tasks:

- Install a new iHub using one of the following options:
  - Automated installation

    Run the installation program to configure iHub and the OOTB PostgreSQL database or an alternative, supported RDBMS.
  - Cloud deployment

    Deploy a prepared image of an installed iHub run-time environment. The administrator deploys the image by unbundling an archive or installing a virtual image on the target machine. The administrator can create a customized image by generating an archive of an installed iHub run-time environment. Alternatively, an out-of-the-box (OOTB) image is available as a separate iHub distribution package in some environments.

- Upgrade an existing Actuate iServer installation from a previous release, such as Release 10 Service Pack 1, to Release 11 Service Pack 4, using the automated installation program. Perform the tasks described in "Performing a side-by-side migration," in Chapter 4, "Migrating to BIRT iHub," to migrate to iHub in a side-by-side installation operation.

For more information on installing iHub, see Chapter 2, "Installing BIRT iHub," later in this book.

Managing the backup, recovery, and failover capabilities of the Encyclopedia volume database and data files

The iHub administrator uses third-party RDBMS tools to manage the backup, recovery, and failover capabilities of the system and Encyclopedia volume database. The iHub administrator uses standard operating system or other third-party tools to manage the backup and recovery of the data files.

The third-party database schemas that contain iHub system and Encyclopedia volume metadata are critical components of BIRT iHub System. To guard against data loss, the database administrator must back up the Encyclopedia volume schemas and all related file data to ensure recoverability in the event of failure. For more information on backing an iHub installation, see Chapter 10, "Backing up an Encyclopedia volume," later in this book.

In iHub and iServer Release 11, it is not necessary to back up the system schema, although future versions may require this operation to protect critical system metadata. The administrator can restore a corrupted or missing system schema using the System Data Store Administrator utility. For more information on this utility, see "Specifying System Data Store Administrator properties" in Chapter 11, "Working with BIRT iHub utilities," later in this book.

An iHub system administrator must take all necessary precautions to ensure that a database is properly backed up and available to safeguard system and Encyclopedia volume metadata. Please consult Actuate Support at the time of installation if you have any questions about the backup, recovery, or failover procedures necessary to protect against the possibility of catastrophic failure.

In Actuate iHub, there is no concept of volume failover, since each node in a cluster can operate on all the volumes. Configuring system and Encyclopedia volume database failover is the responsibility of the third-party RDBMS administrator. The database administrator must use the facilities available in the RDBMS to configure this failover capability. Consult the third-party RDBMS documentation for detailed information on how to use native system tools to configure backup, recovery, and failover operations for the externally managed system and Encyclopedia volume database.

Documentation for a PostgreSQL RDBMS is available at:

```
http://www.postgresql.org/docs/8.4/static/release-8-4.html
```

Documentation for an Oracle RDBMS is available at:

```
http://www.oracle.com/technetwork/database/enterprise-edition
   /documentation/index.html
```

Documentation for Microsoft SQL Server RDBMS is available at:

```
http://msdn.microsoft.com/en-us/sqlserver/bb671149
```

Documentation for IBM DB2 RDBMS is available at:

```
https://www-304.ibm.com/support/docview.wss?uid=swg27009474
```

Managing an iHub cluster

In an iHub system, the concept of a master node no longer exists. Any node in a cluster has the ability to modify the shared server configuration file. The node performing these operations typically depends on which node the system administrator uses when connecting to iHub System through an administration console.

In releases prior to Release 11 Service Pack 1, iServer used multicasting to broadcast event information and synchronize operations in a cluster. Some cloud computing environments do not support multicasting. iHub uses the third-party RDBMS as a shared repository for storing cluster information. This enhancement replaces multicasting as a way of managing cluster information.

Understanding the iHub System process model

The Actuate BIRT iHub System platform uses a multi-threaded, multi-process model, running single instances of the following components on each iHub node:

- Encyclopedia volume
  Stores metadata in an OOTB (PostgreSQL) or alternative RDBMS and coordinates processing for designs, documents, information objects, and other iHub data objects stored in the file system.
- Process Management Daemon (PMD)
  Distributes service requests among available iHub services and nodes.
- iHub servlet container
  Provides the run-time environment for client applications, such as Actuate Information, Management, and Configuration Consoles. Client applications communicate with iHub System using SOAP-based messaging.

In addition, the iHub platform supports multiple instances of the following services on each node:

- Factory
  Executes requests to generate documents and perform server-side printing.
- View
  Supports viewing documents in DHTML and other output formats, such as CSV and PDF. Handles requests to download files from an Encyclopedia volume.
- Integration
  Coordinates the running of data object files that extract data from multiple data sources.

This loosely coupled iHub architecture model provides the following maintenance and performance benefits:

- Startup and shutdown of an iHub node is fast because it is independent of the RDBMS that manages the Encyclopedia volume. The database server can remain online when shutting down an iHub node and is available when the node starts up.
- Controlling the sequence of an Encyclopedia volume startup is not necessary. All volumes are either already online in the database server or come online as the database server starts.
- Downtime to apply a patch or diagnostic fix for an iHub node is reduced. The RDBMS does not have to be shutdown.

Understanding process flow in a stand-alone iHub

Figure 1-1 illustrates iHub process architecture for a stand-alone, two-volume, out-of-the-box (OOTB) PostgreSQL database configuration. In this configuration, the iHub administrator starts and stops an iHub instance by running scripts from the command line or using the graphical user interface (GUI) available in Configuration Console.

The PostgreSQL RDBMS runs as a service in Windows or a process in Linux. The RDBMS can be configured to start automatically or run manually, using a script similar to the iHub startup script.

Client applications, such as Actuate Information, Management, and Configuration Consoles, run in a servlet container. These applications communicate with iHub using the Actuate Information Delivery API or IDAPI.

The Process Management Daemon (PMD) is a message distribution service that routes the request to iHub. iHub receives the request and deserializes the SOAP message. iHub performs the appropriate action and sends a response in the form of a SOAP message back to the application.

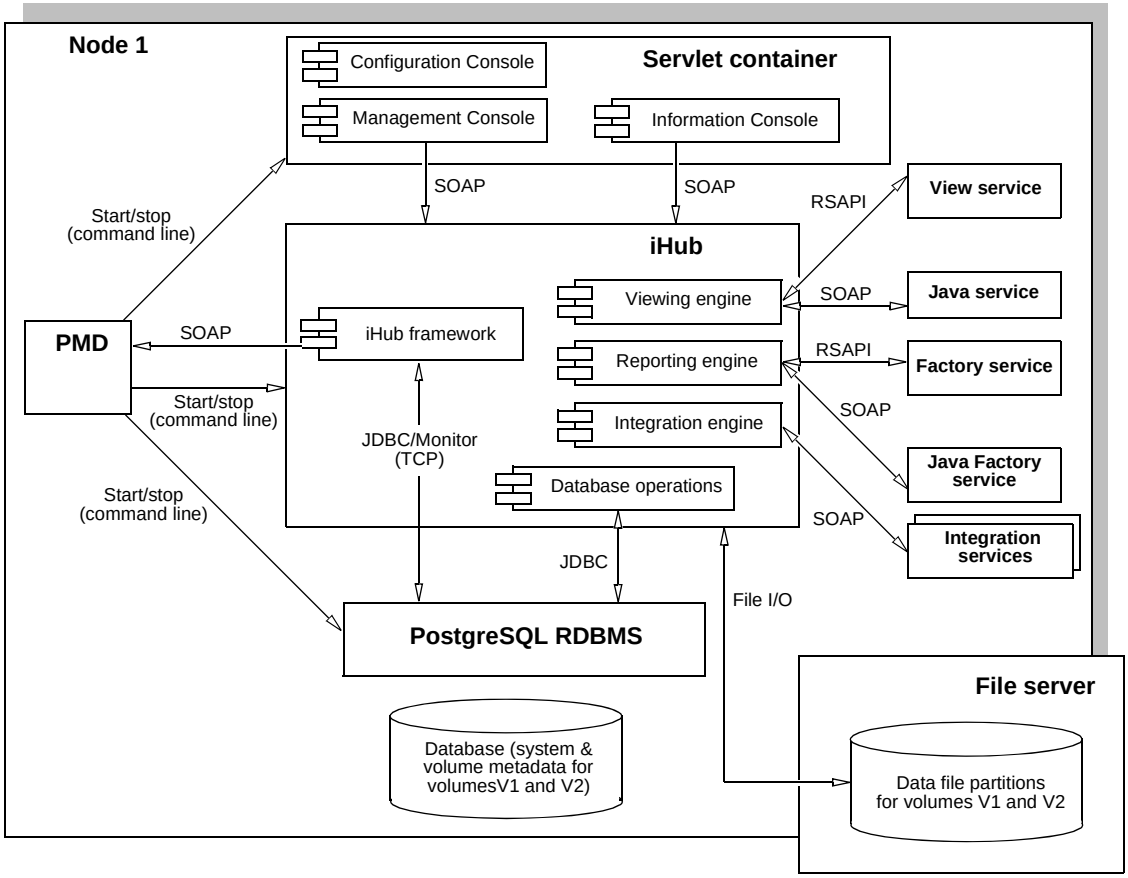


Figure 1-1 iHub process architecture for a stand-alone, two-volume, OOTB database

For example, iHub receives a request to run a design, such as a BIRT design, immediately or as a scheduled job. iHub communicates with the internal iHub framework and Encyclopedia volume metadata database as necessary to locate the design and identify the resources required to run the design in the system.

The reporting engine selects a Java Factory service to run the BIRT design and checks job status. iHub uses an asynchronous Java Factory service to generate a temporary document or a synchronous Java Factory service to generate a scheduled document.

The View service renders the document in DHTML format, or converts the output to other supported formats, such as CSV or PDF, and handles requests to download files from the Encyclopedia volume. The View service sends the document to the requesting application for viewing.

A design that uses a data object utilizes the Integration service to extract data from an external data source.

iHub stores system and Encyclopedia volume metadata in the third-party RDBMS, communicating with the RDBMS as necessary using JDBC. iHub uses the physical file system to read and store designs, documents, and other iHub objects as data in Encyclopedia volume partitions.

The out-of-the-box (OOTB) iHub PostgreSQL installation configures the Encyclopedia volume database on the local disk to increase the reliability and performance of file input and output (I/O) operations. PostgreSQL discourages creating databases accessed using a Network File Systems (NFS) for these reasons. For more information, see section 17.2.1 Network File Systems at the following URL:

`http://www.postgresql.org/docs/8.3/static/creating-cluster.html`

Understanding process flow in an iHub cluster

Figure 1-2 illustrates the iHub process architecture for a clustered, two-node, four-volume, OOTB database configuration. A node is a machine running an iHub instance.

The iHub OOTB PostgreSQL RDBMS starts multiple instances to handle connections for running queries that access metadata. In database jargon, PostgreSQL uses a process-per-user, client/server model. For more information, refer to the PostgreSQL documentation at the following URL:

`http://www.postgresql.org/docs/8.4/static/connect-estab.html`

An iHub administrator adds a node to a cluster to scale iHub System to the necessary processing requirements. There are two methods of adding a node to the cluster:

- Perform an automated, custom installation, using the wizard-driven installation program.
- Perform a manual installation, using the script-driven, cloud deployment package, and a prepared image of an installed iHub run-time environment.

Every cluster node must have network access to the following directory and resources to join the cluster:

- The shared configuration home directory
- Cluster resources, such as printers, database systems, and disk storage systems

Each node gets its configuration from a template in acserverconfig.xml, which is located in a shared configuration home directory along with the license file, acserverlicense.xml.

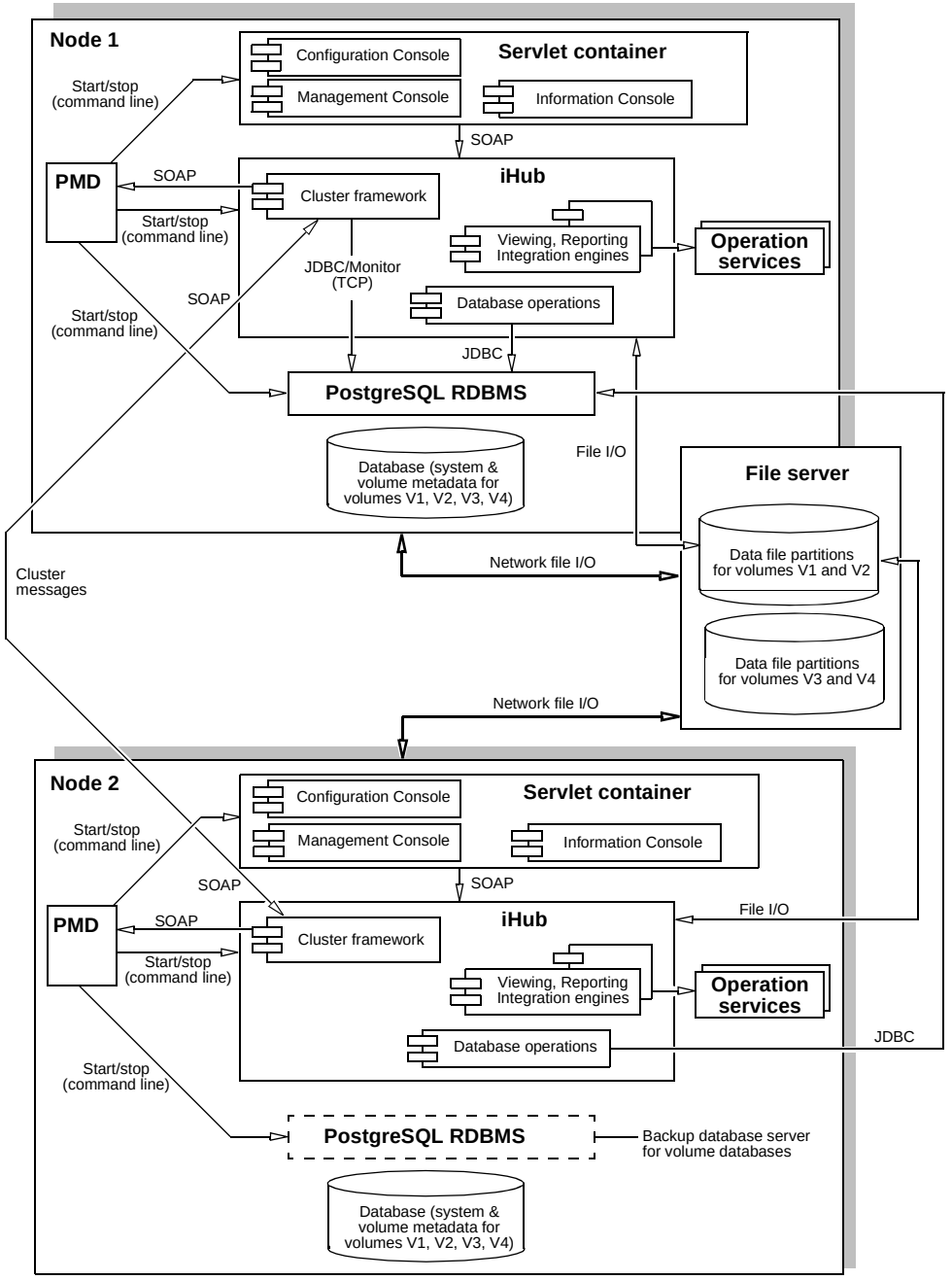


Figure 1-2 iHub RDBMS process architecture for a clustered, two-node, four-volume, OOTB database

The acserverconfig.xml file contains the server templates as well as other configuration parameters specifying the host names, volume names, port numbers, printers, and services used by nodes in the cluster. When the Process Management Daemon (PMD) starts up, it reads these configurations and exposes them to the process environment variable list. When a node joins a cluster, it configures itself using its template.

After installation and configuring the appropriate environment variables in acpmdconfig.xml, the administrator launches the installed iHub image from the command line by passing the necessary arguments or creates a script to execute the command. Nodes with the same cluster ID, running on the same sub-net, automatically detect and join each other to form the cluster. This feature is known as elastic iHub clustering.

The cluster automatically detects the on-off status of any node. Single-point node failure does not affect the availability of other nodes.

In the two-node cluster example, shown in Figure 1-2, client applications, such as Actuate Information, Management, and Configuration Consoles, run in a servlet container. These applications support distributing requests to multiple machines. The cluster communicates across the network using standard HTTP/IP addressing.

One or more nodes in the cluster manage the request message routing. The Process Management Daemons (PMDs) located on each node coordinate processing among available iHub services based on message type to balance load across the nodes.

This loosely coupled model provides the following improvements to intra-cluster messaging:

- Each iHub node in the cluster is relatively independent and identical in terms of components and functionality. Intra-cluster messages are limited to messages for cluster membership and load balancing.
- Operations like design execution and viewing typically require intermediate information from the Encyclopedia volume metadata database. This information is now directly retrieved from or updated in the RBDMS, eliminating internal messages to Encyclopedia services on other nodes.

This increased scalability of operations at the iHub level can create bottlenecks at the RDBMS level. Important factors to consider when configuring nodes and ancillary resources include estimating processing power and access to hardware and software resources, such as printers and database drivers.

iHub instances running on multiple machines maintain iHub system and Encyclopedia volume metadata in a database, which controls access to shared volume data. The volume data can be on machines that are not running iHub, but must be shared and accessible to each iHub instance.

This loosely coupled cluster model provides the following maintenance and performance benefits:

- Startup and shutdown of an iHub node is fast because it is independent of the RDBMS that manages the Encyclopedia volume. An RDBMS can remain online when shutting down an iHub node. The RDBMS is available when the iHub node starts up.
- Controlling the sequence of Encyclopedia volume startup is not necessary. All volumes are either already online or come online as the RDBMS starts.
- Downtime to apply a patch fix patch or a diagnostic fix for an iHub node is reduced. The RDBMS, including the OOTB PostgreSQL database server, does not have to be shutdown. In an iHub cluster, the patch or diagnostic fix can be applied to one iHub node at a time.

This operational model lends itself well to grid, cloud, and other data-center types of deployments.

For more information about the cloud computing deployment option, see Chapter 6, "Installing BIRT iHub in a cloud," later in this book. For more information about the cluster installation option, see Chapter 9, "Clustering," in *Configuring BIRT iHub.*

Administering iHub System

Administering an iHub System includes the following tasks:

- Setting up users, roles, groups, channels, folders, files, and other administrative tasks

  An administrator creates, configures, and manages users, roles, groups, files, folders, and channels, including assigning and updating privileges, managing security role and group memberships, and providing access to channels. User, role, group, and channel privileges selectively control access to the Encyclopedia volume and its data objects.

- Scheduling jobs to run designs and generate documents

  Each stand-alone iHub and node in an iHub cluster has a job scheduler and dispatcher. A job dispatcher send jobs to the local resource group factories.

  In this loosely coupled cluster model, the dispatcher sends a job from the pending queue to available factories, balancing the load across the cluster. Multiple job schedulers running on the nodes in a cluster allow iHub System to scale processing to handle thousands of scheduled jobs at the same time.

- Reviewing logs and auditing the information to diagnose system problems

  iHub can capture usage and error information in log files to assist an administrator in evaluating resource usage and troubleshoot problems. The

usage and error logging applications are open framework applications, which are available as DLLs in Windows and shared libraries in Linux.

- Configuring a cluster using automated installation programs and cloud computing base images

  The administrator can run the installation program to configure iHub or deploy a prepared image of an installed iHub run-time environment. Each cluster node gets its configuration from a template in acserverconfig.xml, located in a shared configuration home directory. Nodes with the same cluster ID, running on the same sub-net, automatically detect and join each other to form the cluster.

- Using Actuate iHub Integration Technology scripts and tools to develop client applications and extend iHub functionality

  The Actuate Information Delivery application programming interface (IDAPI) supports integrating and administering iHub using extensible markup language (XML) and the simple object access protocol (SOAP). Using the IDAPI, developers can create applications that perform such tasks as scheduling a custom event, running an Report Server Security Extension (RSSE) application to manage users and roles in an external system such as an LDAP server, and installing and customizing usage and error logging and performance monitoring extensions.

An iHub administrator uses the Actuate Information, Management, and Configuration Consoles, command-line utilities, and iHub Integration Technology components to perform these tasks.

Please consult the following iHub documentation for more information on how to administer the system using these components:

- *Installing BIRT iHub for Windows* or *Installing BIRT iHub for Linux*

  Describes iHub System architecture. Provides detailed instructions on how to use automated installation programs and command-line utilities to install stand-alone iHub and clustered nodes that store Encyclopedia volume metadata in an external, third-party RDBMS, such as DB2, Oracle, PostgreSQL, or SQL Server. Also describes Actuate licensing policies and procedures and backup and recovery operations.

- *Managing an Encyclopedia Volume*

  Describes how to use Management Console and command-line options to perform tasks such as managing Encyclopedia volume user accounts, assigning privileges, scheduling jobs, and distributing documents.

- *Configuring BIRT iHub*

  Describes how to use Configuration Console to perform tasks such as managing an iHub cluster, adding Encyclopedia volumes to iHub, connecting to a database, updating the license, and configuring iHub properties, such as logging levels, e-mail notification, and printing.

- *Using BIRT iHub Integration Technology*
  Provides information about application programming using the SOAP-based Actuate Information Delivery API (IDAPI), including a Java developer guide and sections on logging, auto archiving, and using the Java Report Server Security Extension (RSSE).

About Migration and Administration Tools

In iHub, the following utilities are obsolete and no longer exist:

- AcExport
  Formerly used to write a copy of the Encyclopedia volume metadata to a file, so the administrator can import the metadata into another release of iHub.

- AcImport
  Formerly used to populate an Encyclopedia volume with metadata previously written to an exported file.

  Upgrade an existing Actuate iServer installation from a previous release, such as Release 10 Service Pack 1, to Release 11 Service Pack 4, then migrate to iHub using a side-by-side installation operation. For more information on installing or migrating to iHub, see the installation and migrating chapters later in this book.

  Back up the database using the utilities that the RDBMS provides. For example, PostgreSQL provides the pg_dump and pg_restore utilities to create and restore a database backup.

  Use operating system or other third-party tools to backup and load designs, documents, and other iHub data objects stored in the file system. For more information on the recommended procedures to back up iHub system and Encyclopedia volume schemas in the iHub environment, refer to Chapter 10, “Backing up an Encyclopedia volume,” later in this book.

- AcToc
  Formerly used to list the contents of an export directory. In iHub, no comparable functionality exists.

- AcVerify
  Formerly used to validate an offline Encyclopedia volume and repair problems.

  In iHub, use the tools available in the third-party RDBMS containing the Encyclopedia volume metadata to verify data integrity and make repairs.

- AcExtern
  Formerly used to convert Encyclopedia volume user security from internal to external registration to allow administration from another system, such as an LDAP server. In iHub, use the iHub Integration Technology custom

application as a reference to configure the Report Server Security Extension (RSSE) when implementing external registration.

- AcIntern

  Formerly used to convert Encyclopedia volume user security from external to internal registration from an LDAP or other system to iHub. In iHub, no comparable functionality exists. For more information on how to install and configure RSSE in the iHub environment, refer to Chapter 11, "Configuring iHub security," in *Configuring BIRT iHub* and Chapter 10, "Using Java Report Server Security Extension" in *Using BIRT iHub Integration Technology.*

- AcMode

  Formerly used to put an Encyclopedia volume in and out of online backup mode. In iHub, a dynamic backup no longer requires putting the system into online backup mode. The administrator performs an Encyclopedia volume metadata backup using the tools provided by the third-party RDBMS, which provides comparable features. The administrator uses standard operating system or other third-party tools to back up the data files.

- AcEncycUpgrade

  Formerly used to convert an older Encyclopedia volume to the latest version. In a manual upgrade process, the administrator uses the Squirrel Data Exporter and Encyclopedia Data Store Administrator utilities to migrate to a new, side-by-side iHub installation. These utilities are Java programs run from the command line.

  For more information on how to migrate to iHub, refer to Chapter 4, "Migrating to BIRT iHub."

Using JDBC to connect to an Encyclopedia volume database

iHub uses JDBC for connecting to the system and Encyclopedia volume database. The iHub run-time JRE environment uses Java 1.6. Any JDBC driver must be compatible with JRE version 1.6 or earlier.

iHub requires a JDBC driver that complies with the JDBC 3.0 specification or later. The function Driver.jdbcCompliant( ) must return TRUE. DatabaseMetadata .getJDBCMajorVersion( ) must return 3 or greater than 3.

An administrator, who decides to customize iHub to connect to a database other than the OOTB PostgreSQL database, must ensure that the JDBC driver returns adequate information about the types on the database. At a minimum, the database must return the common data types, such as integer, floating point, and character. If the database does not return these common data types, then the database administrator must customize the database mapping framework to specify the types.

The JDBC driver must also support the following features:

- Scrollable cursor
- Retention of a cursor after commit
- Update using a prepared cursor

When using connection pooling, the tracing functionality of the JDBC driver captures connection pool run-time statistics.

API Compatibility

iHub provides full backward compatibility with existing applications. Upgrading to an Actuate iHub that utilizes an RDBMS has no impact on any applications that utilize Actuate APIs, such as IDAPI and RSSE.

About international character sets

iHub operates on the assumption that the volume database is configured to run with UTF-8 encoding. Any other database encoding scheme requires configuring the connection parameters to specify the database encoding. The driver must handle the conversion to UCS2.

Administrative reports

The default iHub Encyclopedia volume contains sample BIRT reports that provide information using the metadata and data extracted from the OOTB database, including job schedule, file, and user tracking and usage and error logging. Installing the sample volume is an option in a custom installation.

Supported operating systems

Actuate BIRT iHub supports the following operating systems:

- Windows
- Linux

Part Two

Installing

Chapter

2

Installing BIRT iHub

This chapter discusses the following topics:

- Preparing to install BIRT iHub
- Performing a new installation
- Understanding the iHub installation environment

Preparing to install BIRT iHub

When installing BIRT iHub, the administrator must choose to use the out-of-the-box (OOTB) PostgreSQL relational database management system (RDBMS) or another data store, such as DB2, Microsoft SQL Server, Oracle, or a pre-existing PostgreSQL instance to store Encyclopedia volume metadata. This chapter describes how to install a new instance of BIRT iHub, using the out-of-the-box (OOTB) PostgreSQL RDBMS.

For more information about installing BIRT iHub using an alternative data store, see Chapter 3, "Installing BIRT iHub using an alternative database," later in this book. For more information about upgrading an existing BIRT iHub installation, see Chapter 4, "Migrating to BIRT iHub," later in this book. For information about the BIRT iHub System architecture, see Chapter 1, "Understanding Actuate BIRT iHub architecture," earlier in this book.

Creating an account with Windows administrator privileges

Before installing iHub, create a Windows user account that is a member of the Administrators group. Use this account when installing and running iHub.

The iHub user account must meet the following requirements:

- Be a member of the Windows Administrators group.
  The account must have privileges to access the required software and hardware, such as database servers, printers, and iHub files and folders.
- Have log on as a service privilege.
  If the account does not meet this requirement, the iHub installation program prompts you to configure the privilege to run the Windows Actuate iHub service.

On a new Windows Vista installation, the initial user account is not a member of the Administrators group. You must configure this user account to be a member of this group.

In Windows Server 2003, when installing an iHub that uses the OOTB PostgreSQL RDBMS, create a Windows user account that is a member of the Power Users, not the Administrators group. In Windows Server 2003, PostgreSQL cannot run with Administrator privileges.

Perform the iHub installation using an account that has Administrator privileges. During the installation, when prompted to specify the user account that runs the iHub and PostgreSQL services, specify the Power User account and password. Make sure that this user account has permission to access any required software and hardware, such as database servers, printers, and iHub files and folders.

How to create a Power User account

To create a Power User account, perform the following tasks:

1. In Windows, open the Command Prompt and type:

   ```
   lusrmgr.msc
   ```

2. In Local Users and Groups, choose Users to display the list of users.
3. Double-click the user to display the properties.
4. In Properties—General, deselect Account is disabled, if necessary.
5. In Properties—Member Of, choose Add and perform the following tasks:
   1. On Select Groups, in Enter the object names, type:

      ```
      Power user
      ```

   2. Choose Check Names then choose OK.
6. Exit Local Users and Groups.

In a Windows installation, verify that the ICU_DATA environment variable is set to the location of the Actuate ICU library. The Actuate installation process sets ICU_DATA to the location of the library on the iHub machine. Change this setting, if necessary, to the location of the library. The following example shows the default path set by the iHub installation process:

```
ICU_DATA= C:\WINNT\system32\
```

If you plan to install iHub processes on a machine controlled by a domain server, install iHub while logged into a user account controlled by the local machine, not the domain server. When you create an iHub cluster, all iHub nodes in the cluster must be installed and run under the same user account.

Configuring the iHub user account

Actuate recommends creating a dedicated user account for installing and running iHub. Having a dedicated user account isolates iHub-specific issues and events on a machine, making it easier to administer the environment.

How to configure a user account with administrator privileges

To configure a user account with administrator privileges for installing and running iHub, perform the following tasks:

1. In Windows, open the Command Prompt and type:

   ```
   lusrmgr.msc
   ```

2. In Local Users and Groups, choose Users to display the list of users.
3. Double-click the user to display the properties.
4. In Properties—General, deselect Account is disabled, if necessary.

5 In Properties—Member Of, choose Add and perform the following tasks:

   1 On Select Groups, in Enter the object names, type:

   ```
   Administrators
   ```

   2 Choose Check Names then choose OK.

6 Exit Local Users and Groups.

Configuring log on as a service privilege

The iHub installation program prompts you to configure log on as a service privilege if the iHub user account does not have this privilege.

How to configure the log on as a service privilege manually

To configure the log on as a service privilege manually, perform the following tasks:

1 In Windows Control Panel, open Administrative Tools➤Local Security Policy.

2 In Local Security Settings, navigate to Security Settings➤Local Policies➤User Rights Assignments.

3 In User Rights Assignments, perform the following tasks:

   1 Open Log on as a service Properties. Choose Add User or Group.

   2 In Select Users or Groups, add the user name. Choose Check Names then choose OK.

4 Exit Local Security Settings.

Backing up iHub system and Encyclopedia volume metadata

The third-party database schemas that contain iHub system and Encyclopedia volume metadata are critical components of BIRT iHub System. To guard against data loss, the database administrator must back up the schemas using the tools and resources of the third-party database system.

An iHub system administrator must take all necessary precautions to ensure that the schemas are properly backed up to safeguard the metadata. Please consult Actuate Support at the time of installation if you have any questions about these backup procedures to protect against the possibility of catastrophic failure. For information on the recommended procedures to back up an iHub system and Encyclopedia volume schemas in the Release 11 environment, refer to Chapter 10, “Backing up an Encyclopedia volume,” later in this book.

When installing BIRT iHub, be sure to run the same versions of all products. Upgrade all products at the same time to maintain consistency in the versions you run.

If you are a purchasing customer, you can download iHub from an Actuate ftp site. If you are evaluating BIRT iHub, you can download iHub from BIRT Exchange at the following location:

`http://www.birt-exchange.com`

Actuate also supports the cloud deployment of BIRT iHub using a ready-to-launch iHub image. For more information about this installation option, see Chapter 6, "Installing BIRT iHub in a cloud," later in this book.

The following sections describe how to install a new BIRT iHub using the available installation options.

Performing a new installation

Installing a new BIRT iHub creates a default Encyclopedia volume without migrating data from a pre-existing volume. The default installation program performs the following operations:

- Installs and initializes iHub and the PostgreSQL relational database management system (RDBMS)
- Creates a database in the PostgreSQL RDBMS containing Encyclopedia volume data
- Creates the iserver user in the PostgreSQL RDBMS to access the system
- Creates the system and volume schema, initializing these schema with basic configuration information
- Creates the iHub configuration file, specifying system, volume, and connection information for the default installation

The default installation program also initializes the iserver, system, and volume user passwords to the PostgreSQL superuser password.

Installing a new instance of BIRT iHub

The following section describes how to install a new instance of BIRT iHub in the Windows operating system.

How to perform a new installation of BIRT iHub in Windows

To install iHub, perform the following tasks:

1. Download the iHub distribution package from an FTP software distribution site. Extract the files. Run the self-extracting executable file, ActuateBIRTiHub.exe. The welcome message appears, as shown in Figure 2-1. Choose Next.

Figure 2-1 Viewing the welcome message

2 Read and accept the license agreement, as shown in Figure 2-2. Choose Next.

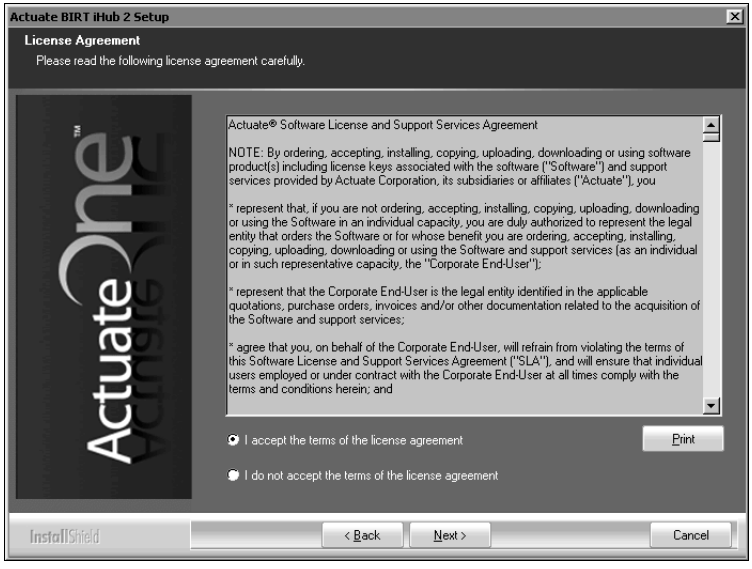


Figure 2-2 Accepting the license agreement

3 The setup installs the prerequisite components that BIRT iHub requires, as shown in Figure 2-3. Choose Next.

Figure 2-3 Installing prerequisites

4 In Setup Type, select Typical to install the default configuration for a stand-alone iHub, as shown in Figure 2-4. Alternatively, choose Custom for one of the following reasons:

- To install individual iHub components
- To install a cluster node
- To install a custom configuration for a stand-alone iHub and a supported relational database management system (RDBMS), such as DB2, Microsoft SQL Server, Oracle, or a pre-existing PostgreSQL instance for the Encyclopedia volume metadata database
- To use an optional Open Security application to control access to iHub using an external system, such as an LDAP server

In Destination Folder, accept the default or choose a new destination folder for the Program and Data locations.

iHub uses the Program location to resolve the paths to all the binaries that it launches. The environment variable, AC_SERVER_HOME, points to the location of the iHub binaries. The default path for the program location is C:/Program Files (x86)/Actuate/iHub2.

iHub uses the Data location to store the iHub logs, iHub Encyclopedia, including the PostgreSQL data, and all other run-time data. The environment variable, AC_DATA_HOME, points to the location of the iHub data. The default path for the data location is C:/Actuate/iHub/data. Choose Next.

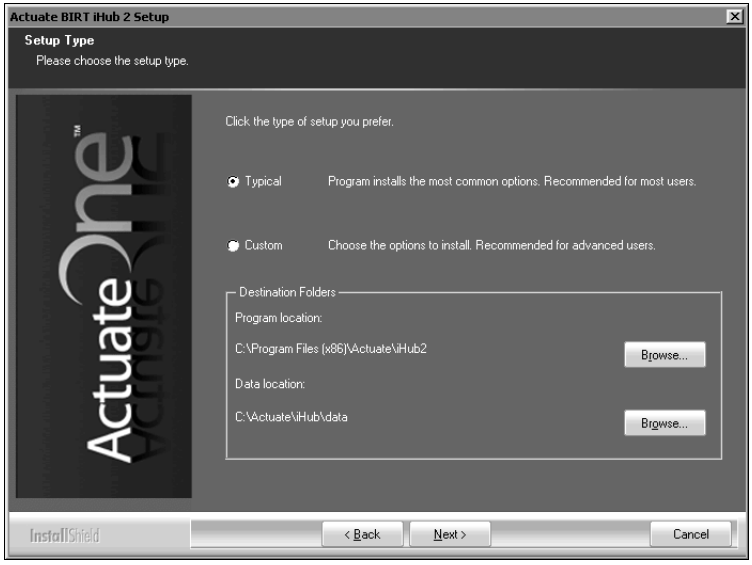


Figure 2-4 Specifying typical or custom setup type

5 In Encyclopedia Metadata Storage and System Name, select the type of Encyclopedia volume metadata database to install, as shown in Figure 2-5. This installation example demonstrates installing the bundled OOTB PostgreSQL database. In System Name, type a name for the BIRT iHub System name. iHub assigns this name to the default Encyclopedia volume. Additionally, iHub inserts this name into the names iHub creates for the Encyclopedia volume schema and the iHub system schema. Choose Next.

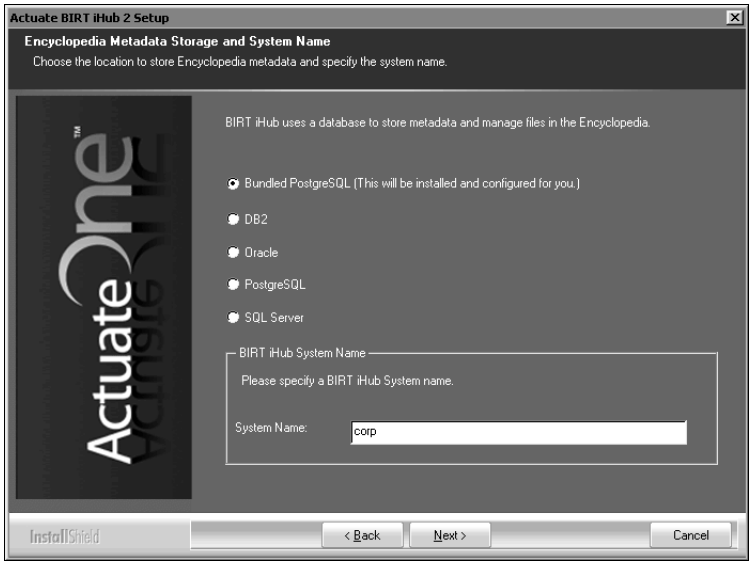


Figure 2-5 Selecting the Encyclopedia volume metadata database to install

6 On PostgreSQL Database Information, in Database Credentials, type and confirm a password, as shown in Figure 2-6. iHub creates the following accounts automatically, using this password for each account:

- postgres
  The PostgreSQL database superuser. The database superuser administers the PostgreSQL relational database management system (RDBMS).
- ac_<BIRT iHub System name>_system
  The System schema owner. iHub creates the iHub system schema and gives it this name. The installation program substitutes <BIRT iHub System name> with the system name you specified in the previous step.
- ac_<BIRT iHub System name>
  The Encyclopedia schema owner. iHub creates the Encyclopedia volume schema using ac_<BIRT iHub System name>.

In Port, accept the default value of 8432. Alternatively, type a different port. Choose Next.

If you receive a message stating that the Database Connection Information port is used by another application, type a different port number, such as 8433.

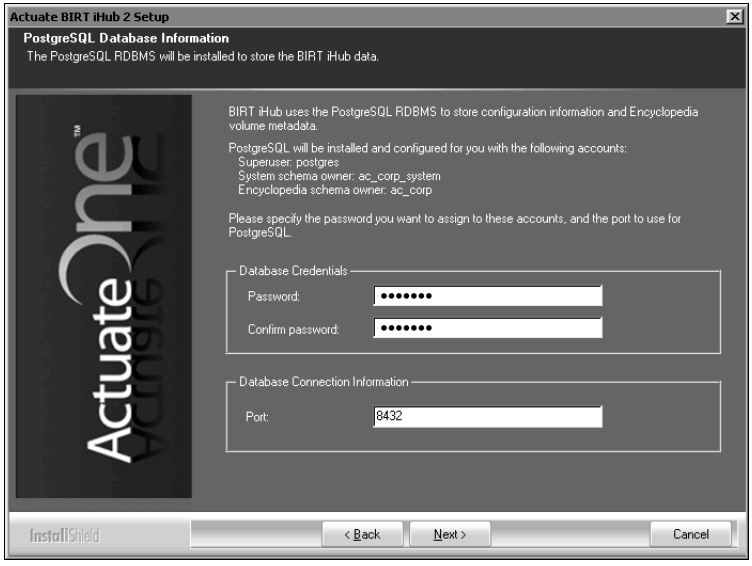


Figure 2-6 Specifying PostgreSQL database information

7 In License File Details, select Use the license that you purchased. Choose Browse then navigate to and choose the license file, as shown in Figure 2-7. Alternatively, choose Try out the product using the included evaluation license if you do not have a purchased license. Choose Next.

Figure 2-7 Specifying the license file

If installing using a named-user license, a prompt appears advising you to check that the volume does not exceed the number of registered users authorized by the license, as shown in Figure 2-8.

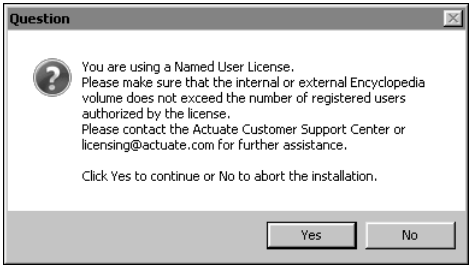


Figure 2-8 Viewing the named-user license question

Choose Yes to continue the installation.

8 In Locale Information, choose Next to accept the default language and time zone, as shown in Figure 2-9.

Alternatively, choose the language and locale settings for your region.

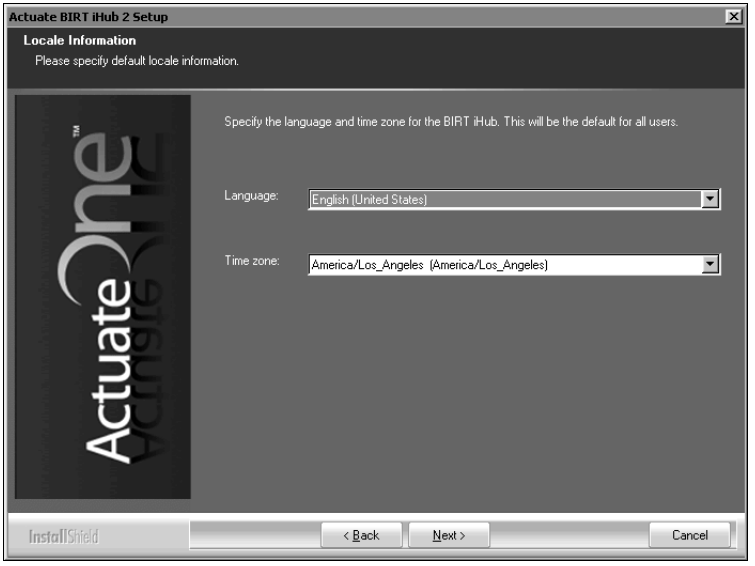


Figure 2-9 Specifying locale information

9 In Specify Profiles, type the user name, password, and confirm the password for the account used to start the BIRT iHub service, as shown in Figure 2-10. The account must be a member of the Administrators group. Actuate recommends that you limit access to this account for security reasons.

If you are installing on Windows 2003, specify a user account that is in the Power Users not the Administrators group. A user account in the Administrators group cannot start the Actuate PostgreSQL for BIRT iHub service.

Accept Automatically start iHub when Windows boots, as shown in Figure 2-10. If you deselect this option, you must start the service manually from Windows Services. Choose Next.

10 In System Configuration Password, type and confirm a password for Configuration Console, as shown in Figure 2-11. For both Configuration Console and Management Console, the default user name is Administrator. The Administrator account for Management Console has no initial password. You can log in to these consoles and change the password settings after installing iHub. Choose Next.

Figure 2-10 Specifying an account for running the iHub service

If prompted to add Log on as a service privilege, choose Yes.

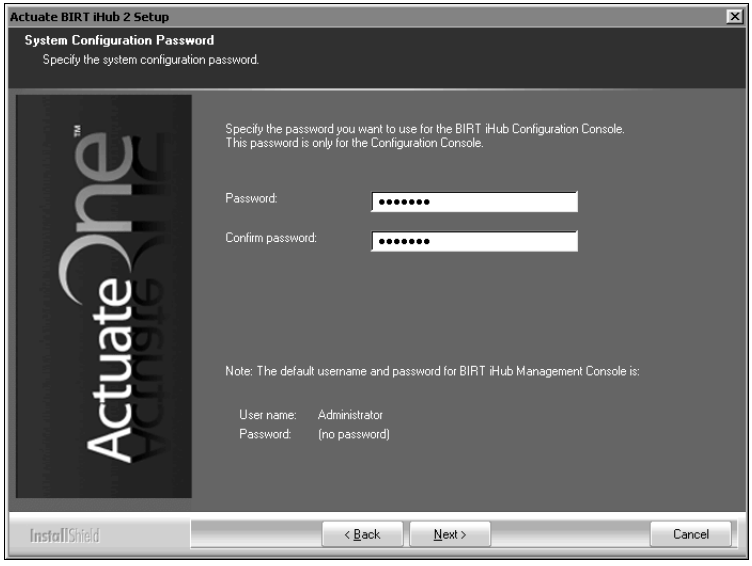


Figure 2-11 Specifying the password for using Configuration Console

11 In Start Copying Files, review the settings shown in Figure 2-12. Choose Next.

Setup Status displays an indicator showing how the installation is progressing, as shown in Figure 2-13.

Figure 2-12 Reviewing settings before copying files

Figure 2-13 Viewing setup status

During installation, a command prompt appears, displaying the run-time commands and messages from the initialization process of the PostgreSQL RDBMS and Encyclopedia volume schema, as shown in Figure 2-14.

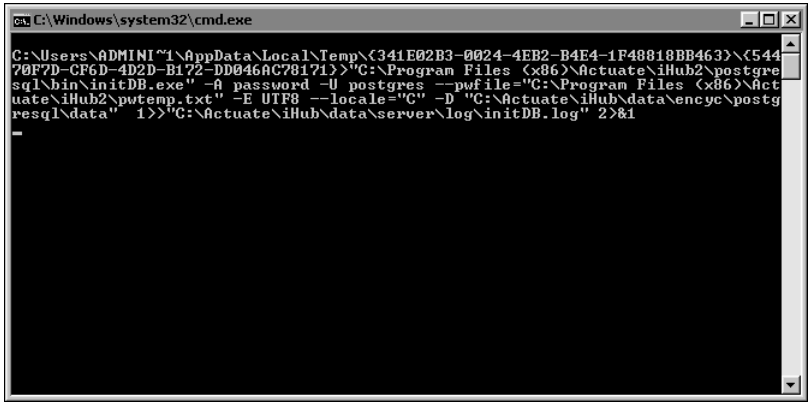


Figure 2-14 Initializing the PostgreSQL RDBMS

A prompt appears, asking if you want to install the pgAdmin database administration tool for the PostgreSQL RDBMS, as shown in Figure 2-15. Choose Yes.

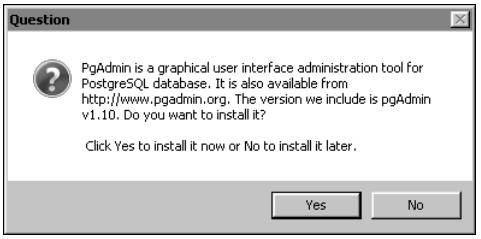


Figure 2-15 Choosing to install the pgAdmin tool

Actuate BIRT iHub 2 Setup appears, then pgAdmin III Setup appears, as shown in Figure 2-16.

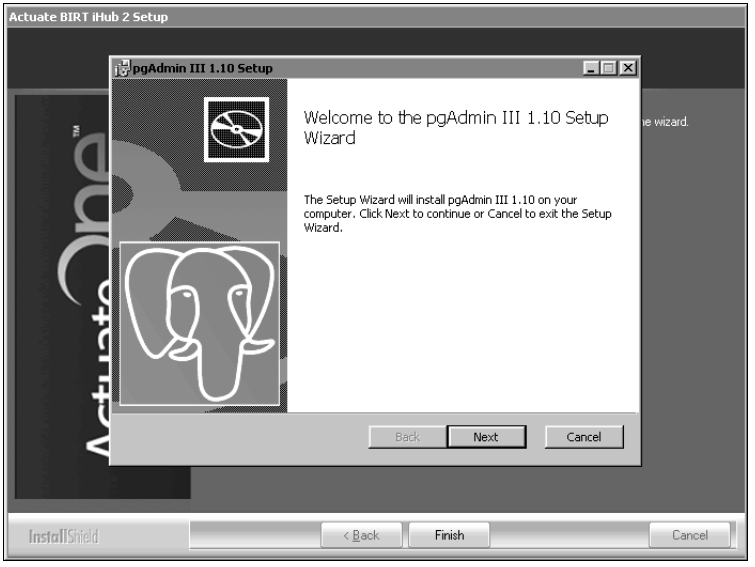


Figure 2-16 Viewing Actuate BIRT iHub and pgAdmin III Setup

12 In pgAdmin III Setup, perform the following tasks:

1. In Welcome, shown in Figure 2-16, choose Next.
2. In End-User License Agreement, select I accept the terms in the License Agreement, as shown in Figure 2-17. Choose Next.

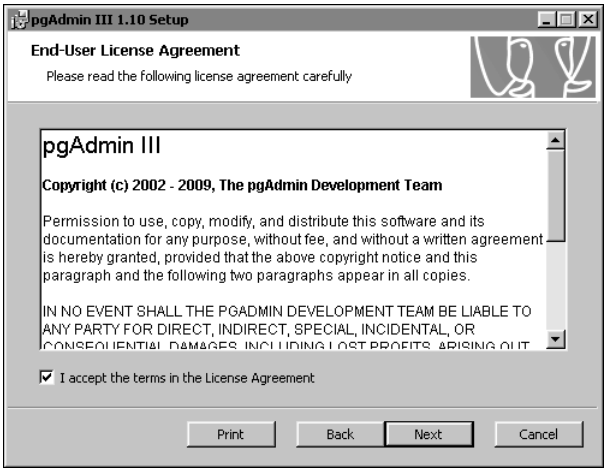


Figure 2-17 Accepting the license agreement

3 In Custom Setup, review the features to be installed, as shown in Figure 2-18. Choose Next.

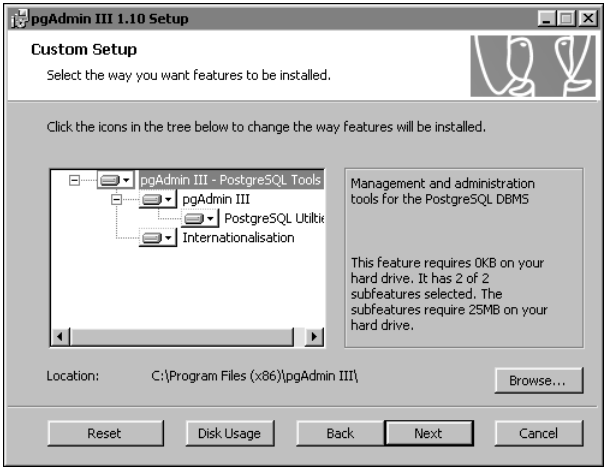


Figure 2-18 Viewing the features to be installed

4 In Ready to Install pgAdmin III, shown in Figure 2-19, choose Install.

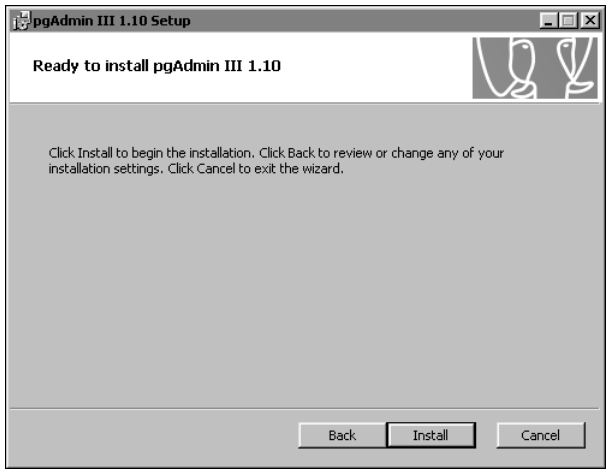


Figure 2-19 Choosing to install pgAdmin III

Installing pgAdmin III appears, as shown in Figure 2-20.

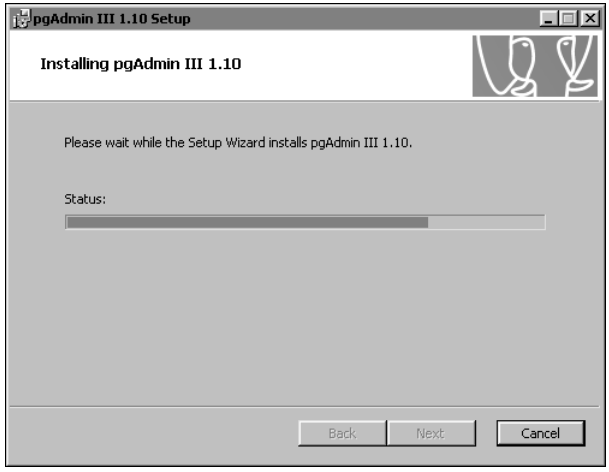


Figure 2-20 Installing pgAdmin III

5 When Completed the pgAdmin III Setup Wizard appears, as shown in Figure 2-21, choose Finish to exit the wizard.

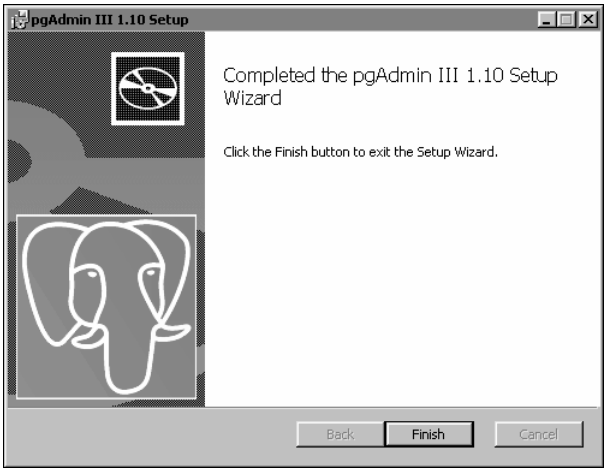


Figure 2-21 Choosing Finish

Choose Finish to exit the wizard, as shown in Figure 2-22.

Figure 2-22 Exiting the installation wizard

13 The installation program prompts you to install the online help from the following location, as shown in Figure 2-23:

`http://www.actuate.com`

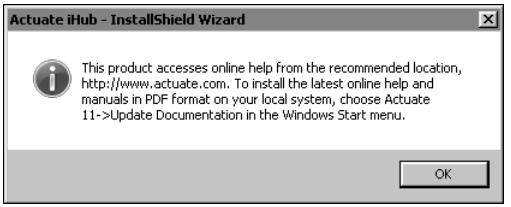


Figure 2-23 Viewing the install online help and manuals prompt

To install the online help and PDF manuals from this location, in Windows, choose Start➤ Actuate➤Update Documentation.

The installation program installs shortcuts on the desktop, as shown in Figure 2-24.

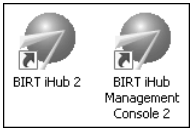


Figure 2-24 Viewing BIRT iHub shortcuts on the desktop

These shortcuts provide access to the following iHub components:

- BIRT iHub Management Console 2
  Launches Management Console to set up user accounts and run designs.
- BIRT iHub 2
  Opens Welcome to Actuate BIRT iHub from which you can log in to Information Console to perform tasks, such as accessing folders and viewing documents.

Accessing the PostgreSQL Database Server using the pgAdmin utility

After migrating the Encyclopedia volume to iHub, you can optionally open the pgAdmin III utility and access the PostgreSQL Database Server to browse the Encyclopedia volume database. Actuate does not support modifying the BIRT iHub PostgreSQL Database schema. Any changes to the schema made by the customer, such as the addition of an index on a table, must be recreated again manually in any future upgrade.

How to access the PostgreSQL Database Server using the pgAdmin utility

To access the PostgreSQL server, choose Start➤Programs➤pgAdmin III 1.10 ➤pgAdmin III.

pgAdmin III appears, showing the PostgreSQL Database Server in the Object browser, as shown in Figure 2-25.

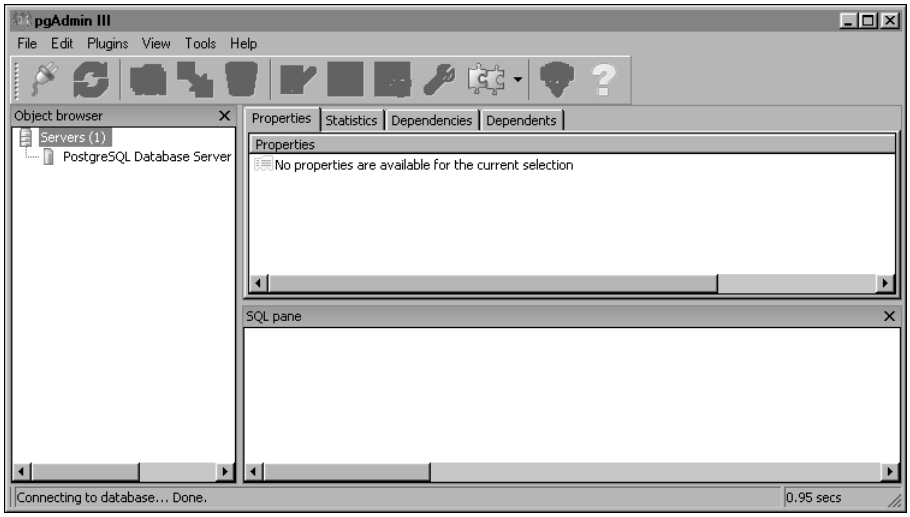


Figure 2-25 Viewing pgAdmin III

If the PostgreSQL Database Server does not appear in the Object browser, you can add the server manually by performing the following tasks:

1 In pgAdmin III, choose File➤Add Server. In New Server Registration—Properties, type or select the following property values, as listed in Table 2-1.

Table 2-1 New server properties

Property	Value
Name	PostgreSQL Database Server
Host	<computer name>
Port	8432
Maintenance DB	postgres
Username	postgres
Password	<your superuser password>
Store password	Not selected
Restore env	Selected
Service	Actuate PostgreSQL for BIRT iHub 2
Connect now	Selected

New Server Registration—Properties appears, as shown in Figure 2-26.

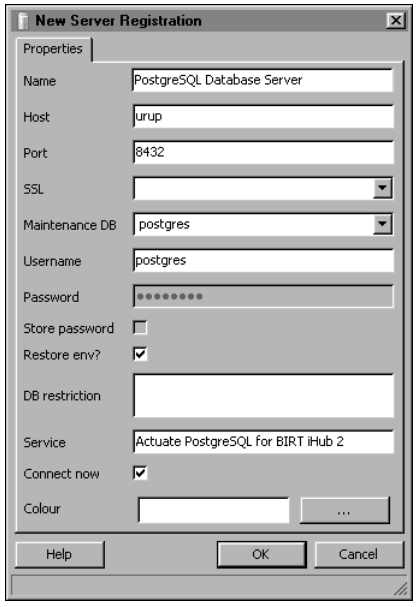


Figure 2-26 Registering a new server

Choose OK. Then, on pgAdmin III, expand PostgreSQL Database Server, as shown in Figure 2-27.

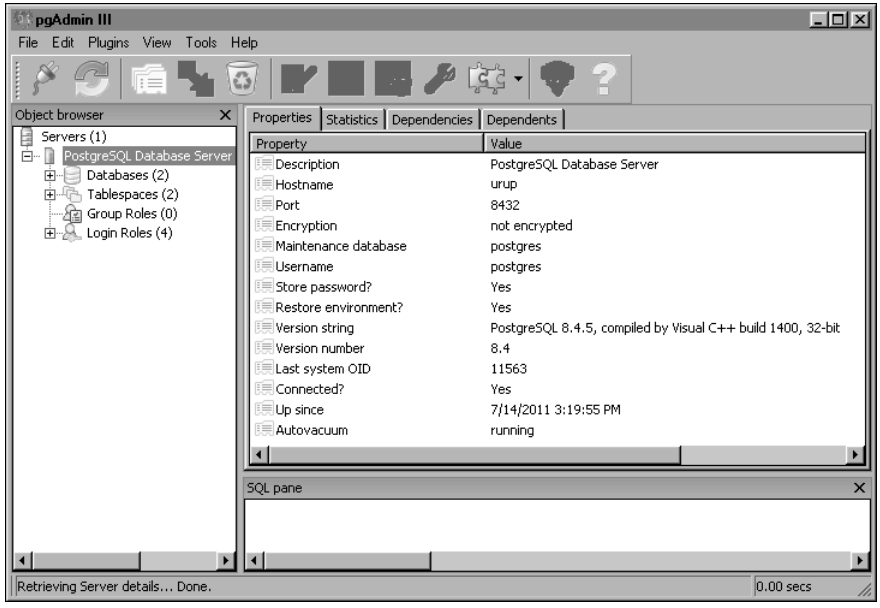


Figure 2-27 Viewing PostgreSQL Database Server properties

2 In Object browser, expand Databases—iserver, iserver—Schemas, Tablespaces, and Login Roles to review the server and database installation, as shown in Figure 2-28.

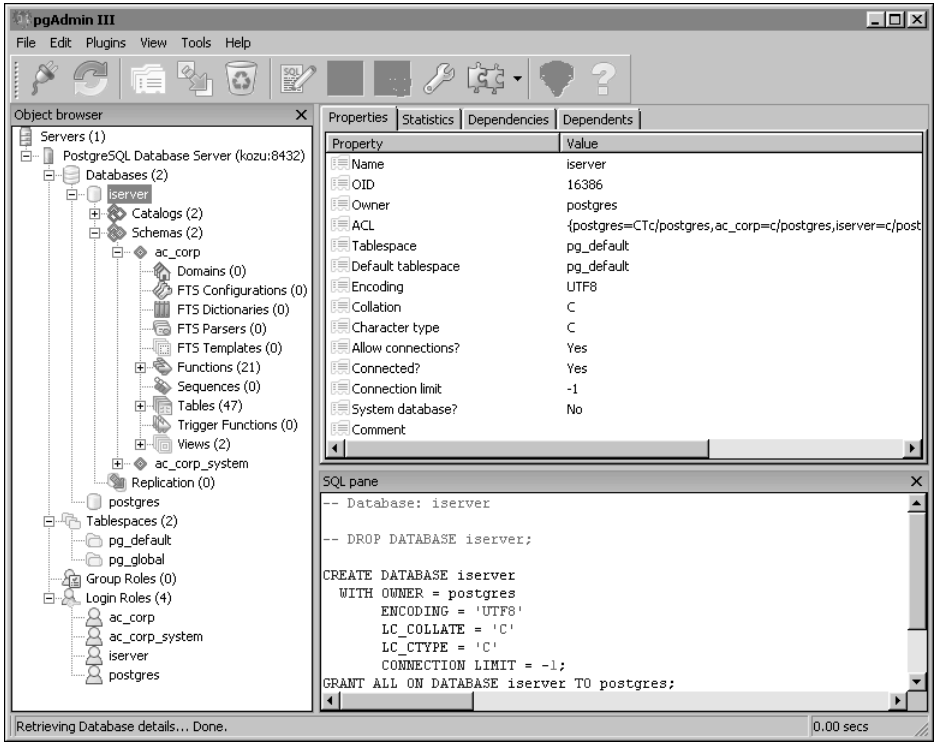


Figure 2-28 Browsing the PostgreSQL Database Server installation

The BIRT iHub uninstall process can optionally remove the iHub installation directory containing the encyc folder, which by default contains the postgres database directory. The uninstall process does not uninstall the pgAdmin III administration tool, a PostgreSQL database not residing in the AC_DATA_HOME/encyc directory, or another third-party database used to store the Encyclopedia volume data. You must uninstall these components separately.

Understanding the iHub installation environment

The following sections provide supplementary information about the iHub installation environment that is useful for an administrator to know.

About migrating an earlier iServer release to Actuate iHub

To upgrade an iServer earlier than Release 8 Service Pack 1, you must first upgrade to Release 11 Service Pack 4, then migrate to iHub by performing a side-by-side installation, as described in Chapter 4, "Migrating to BIRT iHub."

The upgrade program preserves any previous iServer configuration information and reuses the earlier settings. The installation program resolves any differences in default values between releases, ignoring old configuration defaults in favor of new default values. For example, the installation does not prompt the user for port information and machine name. The installation detects the current port numbers and machine name and keeps those settings.

About running different releases on the same machine

Actuate does not support running multiple releases from the same version on a machine. For example, you cannot run iServer Release 11 Service Pack 3 and Release 11 Service Pack 4 on the same machine. Actuate also does not support running BIRT iHub on the same machine as any iServer Release 11 version.

A BIRT iHub can coexist on the same machine with an earlier major release, such as Release 10 Service Pack 1. To run different iServer major releases on the same machine, install the releases in separate directories. Change the default port settings for one release to enable running both versions at the same time.

About performance and disk space issues

During an upgrade installation, the following operations can consume more disk space and take longer than a fresh installation:

- File comparison
- The copy operation to back up original files

During an upgrade, disk space requirements typically double. The installation routine copies files to the local machine for comparison between the original files and the new files. If you perform multiple upgrade installations, the installation routine consumes even more disk space for the backup files you need to restore previous installations.

About the Java Software Development Kit

The iHub installation routine installs the JDK files in:

```
/Program Files (x86)/Common Files/Actuate/22.0/JDK160
```

To use a different JDK with iHub, change the files in the installation directory or change the values of the following environment variables:

- AC_JAVA_HOME
- AC_JVM_HOME
- AC_JRE_HOME
- AC_JRE64_HOME

Using an earlier release of JDK can cause some Actuate features to fail or to work improperly. For example, using an earlier release of JDK can cause Actuate products to display charts incorrectly.

The following types of Actuate executable files use AC_JRE_HOME and AC_JVM_HOME:

- Files containing charts use AC_JVM_HOME to locate the java.exe to generate the chart.
- Files using the Actuate Java Object Interface use AC_JVM_HOME to locate the JVM DLL or library.

Accessing JAR files for document generation

To generate some documents, iHub requires access to jar files in the Jar directory of the iHub installation files. In Windows, include the location of the jar file in the CLASSPATH.

Gathering LDAP information

An optional Open Security application ships with Actuate iHub Integration Technology. This application uses a Lightweight Directory Access Protocol (LDAP) security database to control access to the Encyclopedia volume. To use the Open Security application, you need to perform a custom installation and specify the following additional information:

- Name of the LDAP server and the port on which the LDAP server listens
- LDAP account and password used to query the LDAP server
- LDAP server organization that contains the Actuate users, roles, and groups
- LDAP base domain names and object classes that contain Actuate user, role, and group information
- LDAP group name assigned as the Actuate Encyclopedia volume administrator role

Actuate Open Security uses an LDAP configuration file to map the Encyclopedia volume user information to LDAP object attributes. For more information on

Actuate Open Security, see the reference implementations available in Actuate iHub Integration Technology.

Following best practices

Before deploying a new release in a production environment, Actuate recommends testing the installation.

Using a test environment

Set up a test environment then migrate to iHub when the testing is complete. You cannot mix Actuate products from different release levels. For example, you cannot use Actuate 10 design tools with BIRT iHub.

How and when you upgrade to iHub depends on your site configuration and requirements. Complete the following general tasks in this order to determine how to upgrade your site to iHub:

- Create a test environment for iHub. The test environment can be on the same machine that hosts the earlier Actuate installation or on a separate machine.
- Install the software in the test environment and upgrade earlier versions of designs and files. Also update any custom applications that you built using Actuate iHub Integration Technology. Verify that your applications work properly in the new Actuate iHub Integration Technology test environment.
- Ask application developers and a few users to perform some typical tasks in the test environment.
- Create a production staging area.
- Install the remaining iHub desktop products, if required, in production environments on the user workstations. Verify that the desktop products function properly.
- Schedule a low-impact time to switch to iHub to complete the transition.

Setting up a production staging area

A production staging area is one that you can use for testing and also configure as the live production system. The production staging area can be a separate configuration on the live production machine or a separate machine. You can install all iHub products or the iHub server products and a subset of the desktop products.

If you plan to test iHub desktop products, identify which users to include in the final testing. Developers and users can then confirm that applications perform as expected in the iHub production staging environment.

Complete the following general tasks to test iHub:

- Install BIRT iHub software in a production staging area.
- Install iHub desktop software on the test user machines. Using separate folders, you can install iHub desktop software in conjunction with the earlier desktop software. Users can continue to use the existing Actuate software in production while testing the iHub desktop software.
- Verify that the iHub production staging environment works correctly.
- Install the remaining iHub desktop products, if you installed a subset earlier.
- Verify that all the iHub desktop products work correctly.
- Begin setting up a production environment, described in the following section.

Setting up a production environment

When testing is complete, confirm that your applications work as expected in the iHub environment. Set up the production environment and schedule a date and time to switch from earlier versions to iHub.

When you switch to iHub, use the following procedure list as a general guideline:

- Shut down all Actuate servers.
- Back up earlier Actuate Encyclopedia volumes.
- Upgrade existing Encyclopedia volume schemas. Install upgraded design and document files.
- Install upgraded design and document files.
  Keep Encyclopedia volume data separate from iHub binaries.
- Start BIRT iHub.
- Inform users that they can start using iHub design tool products.

Chapter

3

Installing BIRT iHub using an alternative database

This chapter discusses the following topics:

- Preparing to install BIRT iHub using an alternative database
- Installing an Encyclopedia volume that uses an alternative database

Preparing to install BIRT iHub using an alternative database

When installing BIRT iHub, the administrator must choose to use the out-of-the-box (OOTB) PostgreSQL database or an alternative data store, such as DB2, Microsoft SQL Server, Oracle, or a pre-existing PostgreSQL instance to store Encyclopedia volume metadata. This chapter describes how to install a new instance of BIRT iHub, using an alternative data store.

For all database systems other than OOTB PostgreSQL, the database administrator must create the system and Encyclopedia volume schemas and an iHub application user before installing BIRT iHub. During the iHub installation, the administrator provides the iHub system name, plus the system and Encyclopedia volume schema owner, and iHub application user credentials. The iHub installation program creates the necessary database structures, then loads the metadata.

Creating an account with Windows administrator privileges

Before installing iHub, create a Windows user account that is a member of the Administrators group. Use this account when installing and running iHub.

The iHub user account must meet the following requirements:

- Be a member of the Windows Administrators group.
  The account must have privileges to access the required software and hardware, such as database servers, printers, and iHub files and folders.
- Have log on as a service privilege.
  If the account does not meet this requirement, the iHub installation program prompts you to configure the privilege to run the Windows BIRT iHub service.

On a new Windows Vista installation, the initial user account is not a member of the Administrators group. You must configure this user account to be a member of this group.

Creating the iHub system and Encyclopedia volume schemas and iserver user in an alternative database

Before installing BIRT iHub to use a pre-existing RDBMS, the database administrator must first run SQL scripts that contain the appropriate Data Definition Language (DDL) statements to create a database and the following schema owner and application user accounts with appropriate privileges:

- iHub system schema owner

- Encyclopedia volume schema owner
- iHub application user

Restrict schema and the iHub application user names to alphanumeric and underscore characters with an initial alphabetic character in the pattern [a-z][a-z 0-9]*. Do not use a hyphen.

In an environment containing multiple Encyclopedia volume schemas, Actuate recommends using one iHub application user with privileges on all the schemas. This configuration allows iHub to maximize connection pooling and minimize the number of connections to the RDBMS.

Creating the system and Encyclopedia volume schemas and iserver user in a pre-existing PostgreSQL database

The following SQL scripts provide an example of DDL statements that create the database, schema owners, and an iHub application user role, then grant privileges in a pre-existing PostgreSQL server installation. These steps are not necessary when adding an Encyclopedia volume to an existing schema.

The PostgreSQL database administrator may need to modify these SQL command examples for the specific PostgreSQL installation. In the commands, substitute system and volume schema names appropriate to your environment.

Creating a database

Connect to the PostgreSQL system database as a user with full administrator privileges, typically named postgres, and execute the following SQL commands to create a database named iserver:

```
CREATE DATABASE iserver
   WITH OWNER = "postgres"
   TEMPLATE = template0 ENCODING = 'UTF-8';
REVOKE ALL ON DATABASE iserver FROM PUBLIC;
```

In the iserver database, create the plpgsql procedural language by executing the following SQL command:

```
CREATE LANGUAGE plpgsql;
```

plpgsql is a superset of PostgreSQL SQL that supports advanced programming features, such as variables, conditional expressions, iterative constructs, and events. If the language is already installed, an error message appears. If so, ignore the message.

Creating the system schema owner

In an iHub installation, the system schema owner must have the same name as the system schema. The system schema owner has all privileges on the schema used for the system data store and can grant privileges to other users. The system schema owner must be able to create database objects, such as tables and indexes.

The following commands create a user role named ac_corp_system with appropriate privileges to connect to the previously created iserver database. Connect to the PostgreSQL system database as a user with full administrator privileges and execute the following SQL commands:

```
CREATE ROLE ac_corp_system LOGIN PASSWORD 'password';
GRANT CONNECT ON DATABASE iserver TO ac_corp_system;
```

Creating the Encyclopedia volume schema owner

In an iHub installation, the Encyclopedia volume schema owner must have the same name as the Encyclopedia volume schema. The Encyclopedia volume schema owner has all privileges on the schema used for the Encyclopedia volume data store and can grant privileges to other users. The Encyclopedia volume schema owner must be able to create database objects, such as tables and indexes.

The following commands create a user role named ac_corp with appropriate privileges to connect to the previously created iserver database. Connect to the PostgreSQL system database as a user with full administrator privileges and execute the following SQL commands:

```
CREATE ROLE ac_corp LOGIN PASSWORD 'password';
GRANT CONNECT ON DATABASE iserver TO ac_corp;
```

Creating the iHub application user

iHub connects to the database as an application user. The application user requires only the privileges necessary to perform basic SQL Data Manipulation Language (DML) operations, such as SELECT, INSERT, UPDATE, and DELETE. This user does not require privileges to create or modify the structure of the database.

The following SQL script provides an example of DDL statements that create the iserver user role in a pre-existing PostgreSQL database. Connect to the PostgreSQL system database as a user with full administrator privileges and execute the following SQL commands:

```
CREATE ROLE iserver LOGIN PASSWORD 'password';
GRANT CONNECT ON DATABASE iserver TO iserver;
```

Creating the system schema

The system schema must have the same name as the system schema owner. The following commands create a system schema named ac_corp_system, owned by the user, ac_corp_system, then grant privileges to use that schema to the application user role named iserver. Connect to the iserver application database, not the PostgreSQL system database, as a user with full administrator privileges and execute the following commands:

```
CREATE SCHEMA ac_corp_system AUTHORIZATION ac_corp_system;
GRANT USAGE ON SCHEMA ac_corp_system TO iserver;
```

Creating the Encyclopedia volume schema

In an iHub installation, the Encyclopedia volume schema must have the same name as the Encyclopedia volume schema owner. The following commands create an Encyclopedia volume schema named ac_corp, owned by the user, ac_corp, then grant privileges to use the schema to the application user role named iserver. Connect to the iserver application database, not the PostgreSQL system database, as a user with full administrator privileges and execute the following commands:

```
CREATE SCHEMA ac_corp AUTHORIZATION ac_corp;
GRANT USAGE ON SCHEMA ac_corp TO iserver;
```

Creating the system and Encyclopedia volume schemas and iserver user in a pre-existing Oracle database

In Oracle, there is a one-to-one relationship between a user and a schema. A schema is not a separate entity. The iHub system schema owner has the same name as the system schema. The Encyclopedia volume schema owner also has the same name as the Encyclopedia volume schema.

The following SQL scripts provide an example of DDL statements that create the database, schema owners, and iHub application user, then grant privileges in a pre-existing Oracle database. These steps are not necessary when adding an Encyclopedia volume to an existing schema.

The Oracle database administrator may need to modify these SQL command examples for the specific Oracle installation. In the commands, substitute system and volume schema names appropriate to your environment.

Creating the system schema owner

The iHub system schema owner has all privileges on the schema used for the system data store and can grant privileges to other users. The system schema owner must be able to create database objects, such as tables and indexes.

The following SQL script provides an example of DDL statements that create the iHub system schema owner and grant privileges in a pre-existing Oracle database:

```
DROP USER ac_corp_system CASCADE;
CREATE USER ac_corp_system
   IDENTIFIED BY password
   DEFAULT TABLESPACE USERS
   TEMPORARY TABLESPACE TEMP;
GRANT CREATE TABLE TO ac_corp_system;
GRANT CREATE VIEW TO ac_corp_system;
GRANT CREATE SEQUENCE TO ac_corp_system;
GRANT CREATE TYPE TO ac_corp_system;
GRANT CREATE PROCEDURE TO ac_corp_system;
GRANT CREATE OPERATOR TO ac_corp_system;
```

```
GRANT CREATE TRIGGER TO ac_corp_system;
GRANT CREATE SESSION TO ac_corp_system;
ALTER USER ac_corp_system QUOTA UNLIMITED ON USERS;
COMMIT;
```

Creating the Encyclopedia volume schema owner

The Encyclopedia volume schema owner has all privileges on the schema used for the volume data store and can grant privileges to other users. The Encyclopedia volume schema owner must be able to create database objects, such as tables and indexes.

The following SQL script provides an example of DDL statements that create the Encyclopedia volume schema owner and grant privileges in a pre-existing Oracle database:

```
DROP USER ac_corp CASCADE;
CREATE USER ac_corp
  IDENTIFIED BY password
  DEFAULT TABLESPACE USERS
  TEMPORARY TABLESPACE TEMP;
GRANT CREATE TABLE TO ac_corp;
GRANT CREATE VIEW TO ac_corp;
GRANT CREATE SEQUENCE TO ac_corp;
GRANT CREATE TYPE TO ac_corp;
GRANT CREATE PROCEDURE TO ac_corp;
GRANT CREATE OPERATOR TO ac_corp;
GRANT CREATE TRIGGER TO ac_corp;
GRANT CREATE SESSION TO ac_corp;
ALTER USER ac_corp QUOTA UNLIMITED ON USERS;
COMMIT;
```

Creating the iHub application user

iHub connects to the database as an application user. The application user requires only the privileges necessary to perform basic SQL Data Manipulation Language (DML) operations, such as SELECT, INSERT, UPDATE, and DELETE. This user does not require privileges to create or modify the structure of the database.

The iHub installation process automatically grants the schema privileges required by the application user. The RDBMS database administrator does not have to configure these privileges manually.

The following SQL script provides an example of DDL statements that create the iserver user in a pre-existing Oracle database:

```
DROP USER iserver CASCADE;
CREATE USER iserver
   IDENTIFIED BY password
```

```
   DEFAULT TABLESPACE USERS
   TEMPORARY TABLESPACE TEMP;
GRANT CREATE SESSION TO iserver;
ALTER USER iserver QUOTA UNLIMITED ON USERS;
COMMIT;
```

Creating the system and Encyclopedia volume schemas and iserver user in a pre-existing SQL Server database

The following Transact-SQL scripts provide an example of DDL statements that create the database, schema owners, and iHub application user, then grant privileges in a pre-existing SQL Server database. These steps are not necessary when adding an Encyclopedia volume to an existing schema.

The SQL Server database administrator may need to modify these SQL command examples for the specific SQL Server installation. In the commands, substitute system and schema names appropriate to your environment.

Creating a database

Connect to the SQL Server master database as a user with full system administrator, sysadmin, privileges and execute the following Transact-SQL commands to create a database named iserver:

```
USE master;
GO
CREATE DATABASE iserver
   COLLATE SQL_Latin1_General_CP1_CI_AS;
GO
```

Any database created for iHub processing must use a case-insensitive collation, such as SQL_Latin1_General_CP1_CI_AS. The names of case-insensitive collations typically include the letters, CI.

Creating the system schema owner

In an iHub installation, the system schema owner must have the same name as the system schema. The system schema owner has all privileges on the schema used for the system data store and can grant privileges to other users. The system schema owner must be able to create database objects, such as tables and indexes.

The following commands create a user named ac_corp_system to function as the system schema owner with appropriate privileges to connect to the previously created iserver database. Connect to the iserver database as a user with full administrator privileges and execute the following SQL commands:

```
USE iserver;
CREATE LOGIN ac_corp_system
   WITH PASSWORD = 'password';
```

```
CREATE USER ac_corp_system
   FOR LOGIN ac_corp_system
   WITH DEFAULT_SCHEMA = ac_corp_system;
GO
GRANT CONNECT TO ac_corp_system;
GRANT CREATE TABLE TO ac_corp_system;
GRANT CREATE VIEW TO ac_corp_system;
GRANT CREATE FUNCTION TO ac_corp_system;
GRANT CREATE PROCEDURE TO ac_corp_system;
GO
```

In the SQL Server environment, the default schema does not have to exist when creating the user. The system administrator can create the schema later.

Creating the Encyclopedia volume schema owner

In an iHub installation, the Encyclopedia volume schema owner must have the same name as the Encyclopedia volume schema. The Encyclopedia volume schema owner has all privileges on the schema used for the Encyclopedia data store and can grant privileges to other users. The Encyclopedia schema owner must be able to create database objects, such as tables and indexes.

The following commands create an Encyclopedia volume schema owner named ac_corp with appropriate privileges to connect to the previously created iserver database. Connect to the iserver database as a user with full administrator privileges and execute the following SQL commands:

```
USE iserver;
CREATE LOGIN ac_corp
   WITH PASSWORD = 'password';
CREATE USER ac_corp
   FOR LOGIN ac_corp
   WITH DEFAULT_SCHEMA = ac_corp;
GO
GRANT CONNECT TO ac_corp;
GRANT CREATE TABLE TO ac_corp;
GRANT CREATE VIEW TO ac_corp;
GRANT CREATE FUNCTION TO ac_corp;
GRANT CREATE PROCEDURE TO ac_corp;
GO
```

In the SQL Server environment, the default schema does not have to exist when creating the user. The system administrator can create the schema later.

Creating the iHub application user

iHub connects to the database as an application user. The application user requires only the privileges necessary to perform basic SQL Data Manipulation

Language (DML) operations, such as SELECT, INSERT, UPDATE, and DELETE. This user does not require privileges to create or modify the structure of the database.

The following SQL script provides an example of DDL statements that create the iserver user in a pre-existing SQL Server database. Connect to the iserver database as a user with full administrator privileges and execute the following SQL commands:

```
USE iserver;
CREATE LOGIN iserver WITH PASSWORD = 'password';
CREATE USER iserver FOR LOGIN iserver;
GO
GRANT CONNECT TO iserver;
GO
```

Creating the system schema

The system schema must have the same name as the system schema owner. The following commands create a system schema named ac_corp_system and grant ownership to the user named ac_corp_system. Connect to the iserver application database, not the SQL Server master database, as a user with full administrator privileges and execute the following commands:

```
USE iserver;
GO
CREATE SCHEMA ac_corp_system AUTHORIZATION ac_corp_system;
GO
```

Creating the Encyclopedia volume schema

In an iHub installation, the Encyclopedia volume schema must have the same name as the Encyclopedia volume schema owner. The following commands create an Encyclopedia volume schema named ac_corp and grant ownership to the user named ac_corp. Connect to the iserver application database, not the SQL Server master database, as a user with full administrator privileges and execute the following commands:

```
USE iserver;
GO
CREATE SCHEMA ac_corp AUTHORIZATION ac_corp;
GO
```

Creating the system and Encyclopedia volume schemas and iserver user in a pre-existing DB2 database

The following SQL scripts provide an example of DDL statements that create the database, schema owners, and iHub application user, then grant privileges in a pre-existing DB2 database. These steps are not necessary when adding an Encyclopedia volume to an existing schema.

The DB2 database administrator (DBA) may need to modify these SQL command examples for the specific DB2 installation. In the commands, substitute system and schema names appropriate to your environment.

Creating user accounts

DB2 uses operating system accounts instead of internally defined database users. A database user must exist as an operating system user account, using the native security mechanisms that the operating system provides, before a user can be referenced in a DB2 system. Once a user exists in the operating system, the DB2 system administrator can assign privileges to that user using DDL statements.

Creating a database

Actuate requires a DB2 database to support VARGRAPHIC columns. DB2 does not support UCS-2 or UTF-16 as the primary encodings for a database. DB2 also sizes the VARCHAR data type in bytes, not characters. To work around these issues, iHub uses VARGRAPHIC instead of VARCHAR. VARGRAPHIC stores UTF-16 data and sizes this data in characters.

Actuate also requires a DB2 database to use a case-insensitive collation, such as UCA500R1_LEN_S2, which is not the default. DB2 supports this functionality only in DB2 9.5 Fix Pack 1 and later versions.

To create the iserver database, connect to the DB2 system as a user with full administrator privileges and execute the following SQL command:

```
CREATE DATABASE iserver
   AUTOMATIC STORAGE YES
   USING CODESET UTF-8
   TERRITORY US
   COLLATE USING UCA500R1_LEN_S2
   PAGESIZE 8192
```

Creating the system schema owner

The iHub system schema owner has all privileges on the schema used for the system data store and can grant privileges to other users. The system schema owner must be able to create database objects, such as tables and indexes.

To create the iHub system schema owner and grant privileges in a pre-existing DB2 database, execute the following command:

```
GRANT CONNECT, LOAD ON DATABASE TO USER ac_corp_system;
```

Creating the Encyclopedia volume schema owner

The Encyclopedia volume schema owner has all privileges on the schema used for the volume data store and can grant privileges to other users. The Encyclopedia schema owner must be able to create database objects, such as tables and indexes.

To create the Encyclopedia volume schema owner and grant privileges in a pre-existing DB2 database, execute the following command:

```
GRANT CONNECT, LOAD ON DATABASE TO USER ac_corp;
```

Creating the iHub application user

iHub connects to the database as an application user. The application user requires only the privileges necessary to perform basic SQL Data Manipulation Language (DML) operations, such as SELECT, INSERT, UPDATE, and DELETE. This user does not require privileges to create or modify the structure of the database.

The iHub installation process automatically grants the schema privileges required by the application user. The RDBMS database administrator does not have to configure these privileges manually.

To create the iserver user in a pre-existing DB2 database, execute the following command:

```
GRANT CONNECT ON DATABASE TO USER iserver;
```

Creating the system schema

The system schema must have the same name as the system schema owner. To create a system schema named ac_corp_system and grant ownership to the user named ac_corp_system, execute the following command:

```
CREATE SCHEMA ac_corp_system AUTHORIZATION ac_corp_system;
```

Creating the Encyclopedia volume schema

In an iHub installation, the Encyclopedia volume schema must have the same name as the Encyclopedia volume schema owner. To create an Encyclopedia volume schema named ac_corp and grant ownership to the user named ac_corp, execute the following command:

```
CREATE SCHEMA ac_corp AUTHORIZATION ac_corp;
```

Adding support for the digit wildcard character in iHub when the metadata database is DB2

In a stand-alone iHub installation that uses DB2 to store Encyclopedia volume metadata, the pound sign ('#') in iHub is treated as a single alphanumeric wildcard character instead of a digit wildcard character. This limitation exists because DB2 does not support regular expressions in SQL syntax.

The administrator can add regular expression testing capability to DB2 by creating a User Defined Function, or UDF. The following article documents this approach:

```
http://www.ibm.com/developerworks/data/library/techarticle
   /dm-1011db2luwpatternmatch/index.html
```

The solution consists of the following parts:

- Create a Java implementation that performs the regular-expression testing functionality. In the article, IBM provides the sample java code for such an implementation. See the Implementation section in the article. IBM also provides the pre-built java package, db2_regex.jar, for download. See the Download section in the article.
- Create a UDF to access the external Java method. Use the following sample implementation as an example:

  ```
  CREATE OR REPLACE FUNCTION REGEXP_LIKE(SOURCE VARCHAR(3000),
     REGEX VARCHAR(512),
  MODE VARCHAR(3))
  RETURNS INTEGER
  FENCED
  NOT DETERMINISTIC
  NO SQL
  LANGUAGE JAVA
  PARAMETER STYLE JAVA
  EXTERNAL NAME
     'db2_regex:com.ibm.avalanche.udf.regex.Regexp.regexpLike'
  NO EXTERNAL ACTION
  @
  ```

  You can create this method in a schema that the Encyclopedia volume schema owner accesses.

  After creating the UDF, grant execute privileges to the volume schema user as well as to the iHub application user by executing the following DDL statements:

  ```
  GRANT EXECUTE ON FUNCTION REGEXP_LIKE( VARCHAR(3000),
     VARCHAR(512),
  VARCHAR(3) )
  TO USER /* volume schema user */ WITH GRANT OPTION

  GRANT EXECUTE ON FUNCTION REGEXP_LIKE( VARCHAR(3000),
     VARCHAR(512),
  VARCHAR(3) )
  TO USER /* iHub application user */
  ```

To support the Encyclopedia engine smart search capability to use this regular expression UDF, update the DB2.xml file in the AC_SERVER_HOME\etc \DataStores\DatabaseConfiguration folder. Change the MatchOpMapper section from the following:

```
<MatchOpMapper SingleMatch="_"
                   GreedyMatch="%"
                   DigitMatch="_"
                   EscapeTemplate="@$"
                   AdditionalSpecialChars="@">
          <FunctionMappings>
                   <FunctionMapping FunctionName="MATCH">
                           $P0 LIKE $P1 ESCAPE '@'
                   </FunctionMapping>
          </FunctionMappings>
</MatchOpMapper>
```

to the following:

```
<MatchOpMapper SingleMatch="."
                   GreedyMatch=".*"
                   DigitMatch="[0-9]"
                   AdditionalSpecialChars="\^.$|()[]*+?{},">
          <FunctionMappings>
                     <FunctionMapping FunctionName="MATCH">
                          REGEXP_LIKE
                          ( $P0 , '^' || $P1 ||  '$$', 'c' ) > 0
                     </FunctionMapping>
          </FunctionMappings>
</MatchOpMapper>
```

Note that you may need to prefix the UDF REGEXP_LIKE with the schema name, depending on where the function is located.

Filtering for a user name returns an empty result if name contains certain special characters

In Management Console—Users, if iHub uses a PostgreSQL database to store metadata, filtering on a user name returns an empty result when certain special characters are used in the filter string. For example, if you filter on a user name containing one or more of the letters, é, à, è, ü, ä, ö, ê, or ô, iHub does not find the name, using following default locale collation and type settings:

```
CREATE DATABASE iserver
   WITH OWNER = postgres
   ENCODING = 'UTF8'
   LC_COLLATE = 'C'
   LC_CTYPE = 'C'
   CONNECTION LIMIT = -1;
```

To resolve this problem, perform the following tasks:

1. Export the iserver database to a dump file.
2. Create a new database named iserver. If you are running iHub on a Windows machine, execute the following DDL statements to recreate the database:

```
CREATE DATABASE iserver
   WITH OWNER = "postgres"
   TEMPLATE = template0 ENCODING = 'UTF-8'
   LC_COLLATE = 'English, United States'
   LC_CTYPE = 'English, United States'
   CONNECTION LIMIT = -1;
```

   If you are running iHub on a Linux machine, execute the following DDL statements to recreate the database:

```
CREATE DATABASE iserver
   WITH OWNER = "postgres"
   TEMPLATE = template0 ENCODING = 'UTF-8'
   TABLESPACE = pg_default
   LC_COLLATE = 'en_US.UTF8'
   LC_CTYPE = 'en_US.UTF8'
   CONNECTION LIMIT = -1;
```

3. Import the data back into the newly created iserver database.

Backing up iHub system and Encyclopedia volume metadata

The third-party database schemas that contain iHub system and Encyclopedia volume metadata are critical components of BIRT iHub System. To guard against data loss, the database administrator must back up the schemas using the tools and resources of the third-party database system.

An iHub system administrator must take all necessary precautions to ensure that the schemas are properly backed up to safeguard the metadata. Please consult Actuate Support at the time of installation if you have any questions about these backup procedures to protect against the possibility of catastrophic failure. For information on the recommended procedures to back up an iHub system and Encyclopedia volume schemas in the iHub environment, see Chapter 10, “Backing up an Encyclopedia volume,” later in this book.

When installing BIRT iHub, be sure to run the same versions of all products. Upgrade all products at the same time to maintain consistency in the versions you run.

If you are a purchasing customer, you can download iHub from an Actuate FTP software distribution site. If you are evaluating BIRT iHub, you can download iHub from BIRT Exchange at the following location:

`http://www.birt-exchange.com`

Actuate also supports the cloud deployment of BIRT iHub using a ready-to-launch iHub image. For more information about this installation option, see Chapter 6, "Installing BIRT iHub in a cloud," later in this book. For information about the BIRT iHub System architecture, see Chapter 1, "Understanding Actuate BIRT iHub architecture," earlier in this book.

The following section describes how to install BIRT iHub using PostgreSQL as an alternative data store.

Installing an Encyclopedia volume that uses an alternative database

The following procedures use a pre-existing PostgreSQL database and schema as an example. During the iHub installation, the administrator provides the schema owner and database user credentials. The iHub installation program creates the necessary volume database structures, then loads the metadata.

How to install an Encyclopedia volume that uses an alternative database

1 Download the iHub distribution package from an FTP software distribution site. Extract the files. Run the self-extracting executable file, ActuateBIRTiHub.exe. The welcome message appears, as shown in Figure 3-1. Choose Next.

Figure 3-1 Viewing the welcome message

2 Read and accept the license agreement, as shown in Figure 3-2. Choose Next.

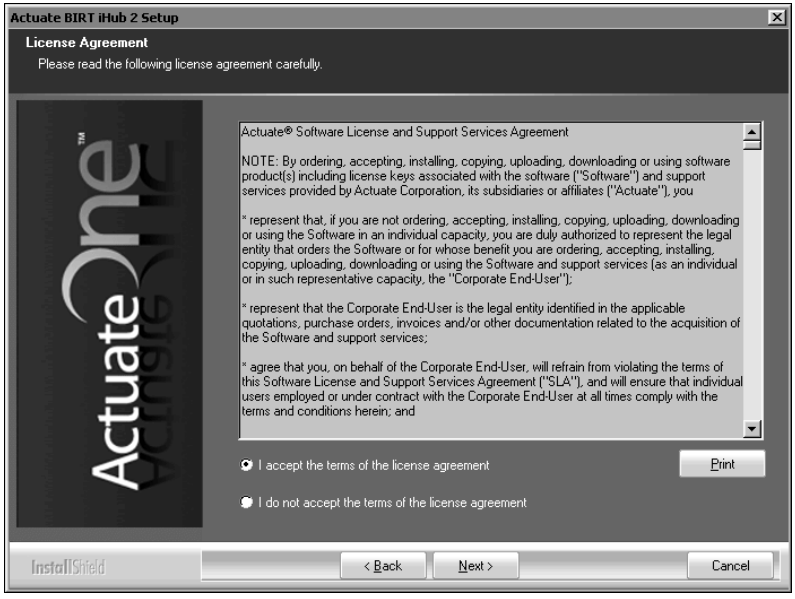


Figure 3-2 Accepting the license agreement

3 The setup installs the prerequisite components that BIRT iHub requires, as shown in Figure 3-3. Choose Next.

Figure 3-3 Installing prerequisites

4 In Setup Type, as shown in Figure 3-4, choose Typical to install a supported relational database management system (RDBMS) for the Encyclopedia volume, such as OOTB PostgreSQL or a pre-existing DB2, Microsoft SQL Server, Oracle or PostgreSQL RDBMS.

In Destination Folder, accept the default or choose a new destination folder for the Program and Data locations.

iHub uses the Program location to resolve the paths to all the binaries that it launches. The environment variable, AC_SERVER_HOME, points to the location of the iHub binaries. The default path for the program location is C:/Program Files (x86)/Actuate/iHub2.

iHub uses the Data location to store the iHub logs, Encyclopedia volume data, and other related data. The environment variable, AC_DATA_HOME, points to the iHub data location. The default path is C:/Actuate/iHub/data.

Choose Next.

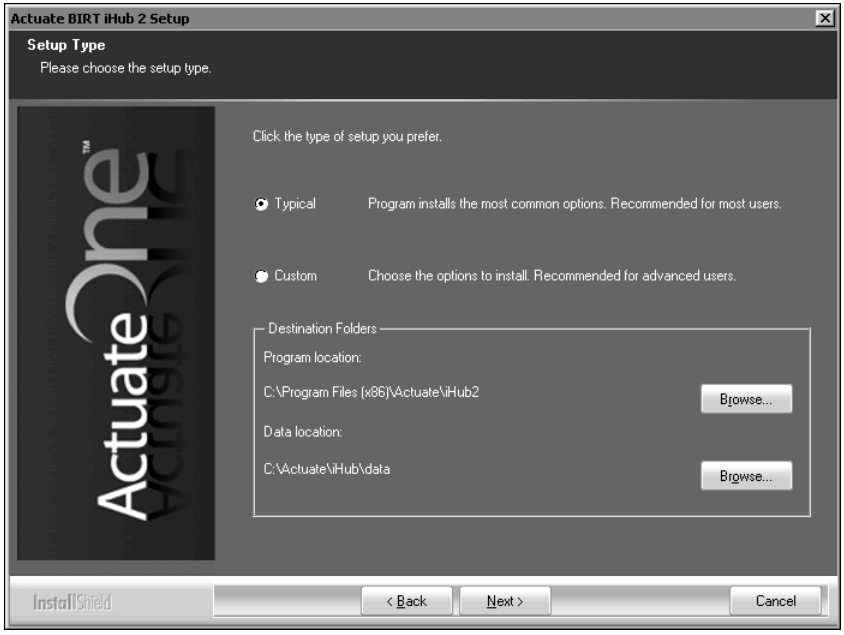


Figure 3-4 Specifying Typical setup type

5 In Encyclopedia Metadata Storage, select DB2, Oracle, PostgreSQL, or SQL Server to use as a pre-existing third-party database for storing Encyclopedia volume metadata. This example uses the PostgreSQL RDBMS, as shown in Figure 3-5. In System Name, type a name for the BIRT iHub system. Restrict the name to alphanumeric and underscore characters with an initial alphabetic character in the pattern [a-z][a-z 0-9]*. Do not use a hyphen.

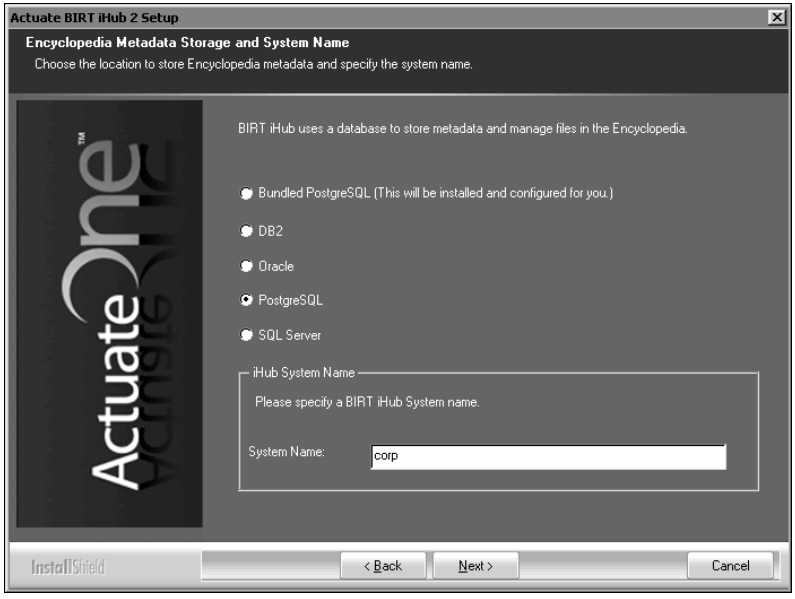


Figure 3-5 Choosing a third-party database

6 After you chose DB2, Oracle, PostgreSQL, or SQL Server in the previous step, 3rd Party Database Information appears. Depending on which alternative database you selected, provide the following database information. Then, choose Next:

- DB2

  For a pre-existing DB2 installation, type the server hostname. In Database, specify the name of the database, such as iserver. Type the port. By default, the port is 50000. You can leave IANAAppCodePage blank. In Database User credentials, specify the iserver application user and a password, as shown in Figure 3-6.

- Oracle

  For a pre-existing Oracle installation, type the server hostname and port. By default, the port is 1521.

  In Service Name, type a valid service name, such as orcl.actuate.com, that identifies the Oracle database server on which you want to install the Encyclopedia volume metadata. Do not use just the system identifier (SID). Provide the complete reference to the server, including the domain. When using a service name, leave TNS Server Name and TNS Names File left blank. When using a Transparent Network Substrate (TNS) service, leave service name blank.

  In Database User Credentials, specify the iserver application user and a password, as shown in Figure 3-7.

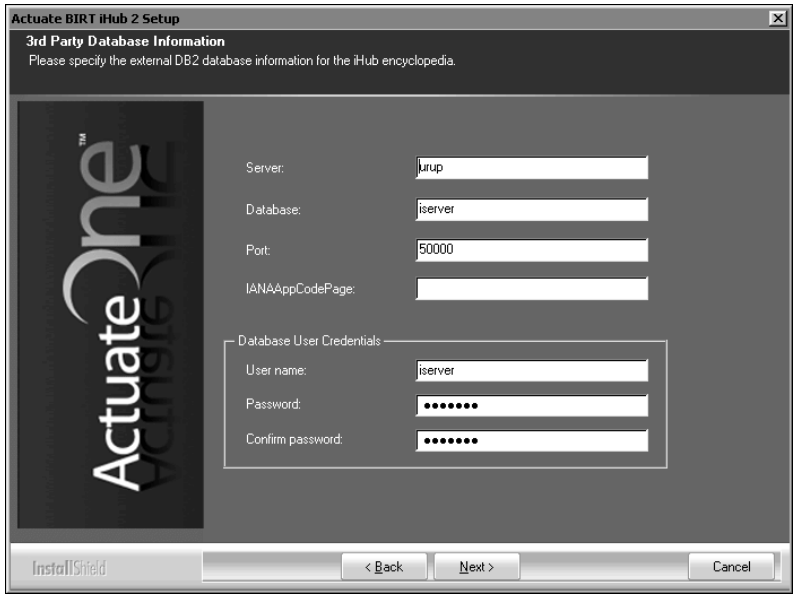


Figure 3-6 Specifying third-party database information for DB2

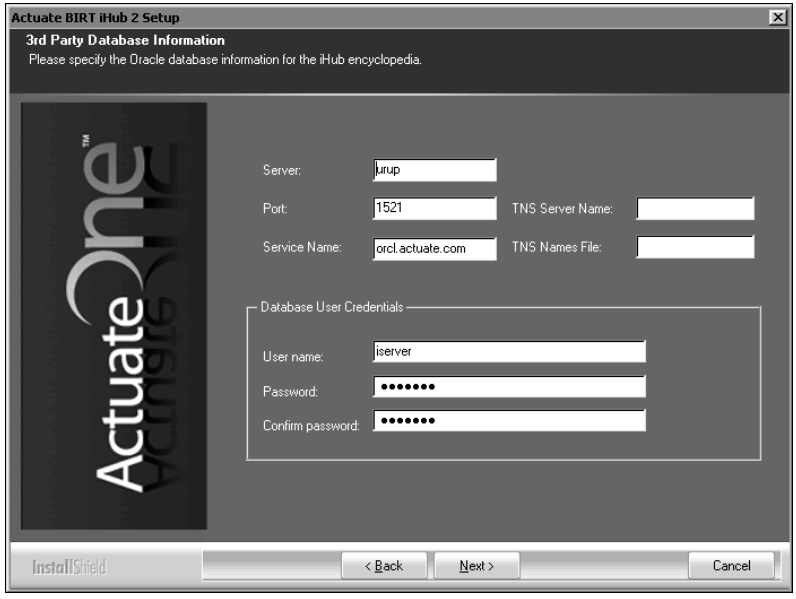


Figure 3-7 Specifying third-party database information for Oracle

- PostgreSQL

  For a pre-existing PostgreSQL installation, type the server hostname and port. By default, the port is 5432. In Database, specify the name of the database, such as iserver. In Database User credentials, specify the iserver application user and a password, as shown in Figure 3-8.

- SQL Server

  For a pre-existing SQL Server installation, type the server hostname and port. By default, the port is 1433. In Database, specify the name of the database, such as iserver. In Instance, type the SQL Server instance name. In Figure 3-9, Instance contains the SQL Server default instance name. Specify the iserver application user in Database User credentials, as shown in Figure 3-9.

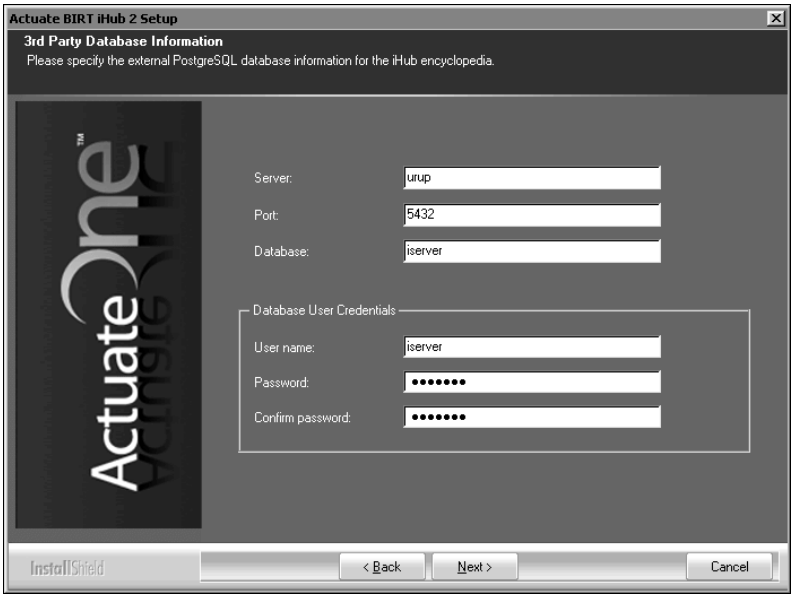


Figure 3-8 Specifying third-party database information for PostgreSQL

7 In Database Schema Information, type the system schema owner, password, and confirm the password in System Database Schema Credentials. Type the Encyclopedia database schema owner, password, and confirm the password in Encyclopedia Database Schema Credentials, as shown in Figure 3-10.

Figure 3-9 Specifying third-party database information for SQL Server

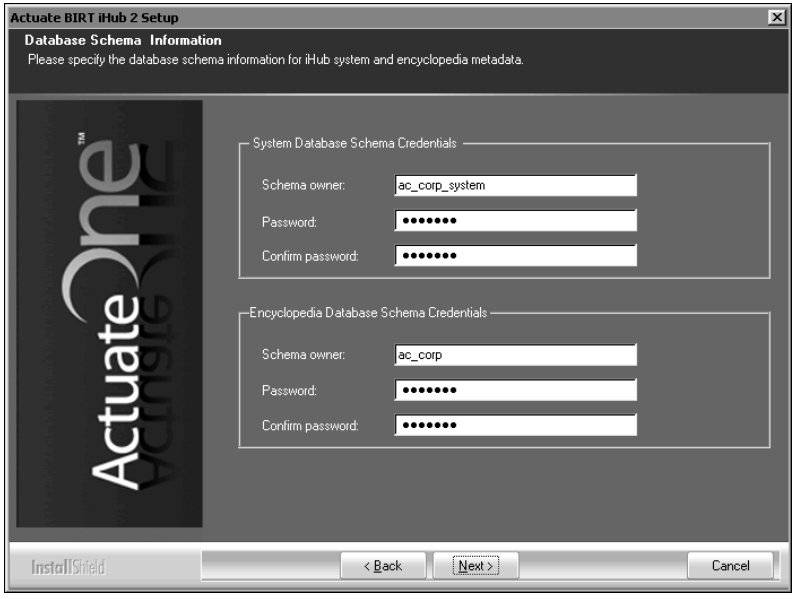


Figure 3-10 Specifying System and Encyclopedia schema passwords

8 In License File Details, select Use the license that you purchased. Choose Browse then navigate to and choose the license file, as shown in Figure 3-11. Choose Next.

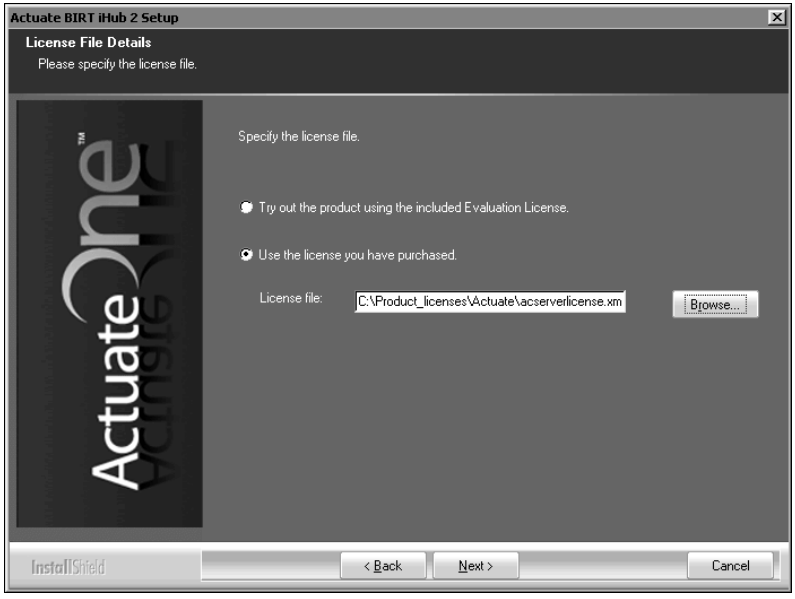


Figure 3-11 Specifying the license file

When installing using a named user license, a prompt appears advising you to check that the volume does not exceed the number of registered users authorized by the license, as shown in Figure 3-12.

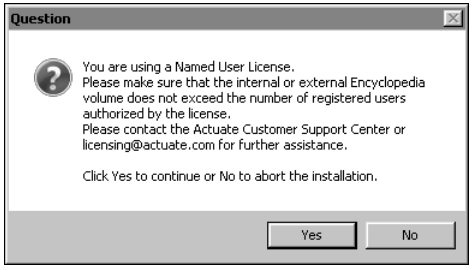


Figure 3-12 Viewing the named user license question

Choose Yes to continue the installation.

9 In Locale Information, choose Next to accept the default language and time zone, as shown in Figure 3-13.

Alternatively, choose the language and locale settings for your region.

10 In Specify Profiles, type the user name, password, and confirm the password for the account used to start the Actuate BIRT iHub service, as shown in Figure 3-14. The account must be a member of the Administrators group. Actuate recommends that you limit access to this account for security reasons.

Accept Automatically start iHub when Windows boots, as shown in Figure 3-14. If you deselect this option, you must start the service manually from Windows Services. Choose Next.

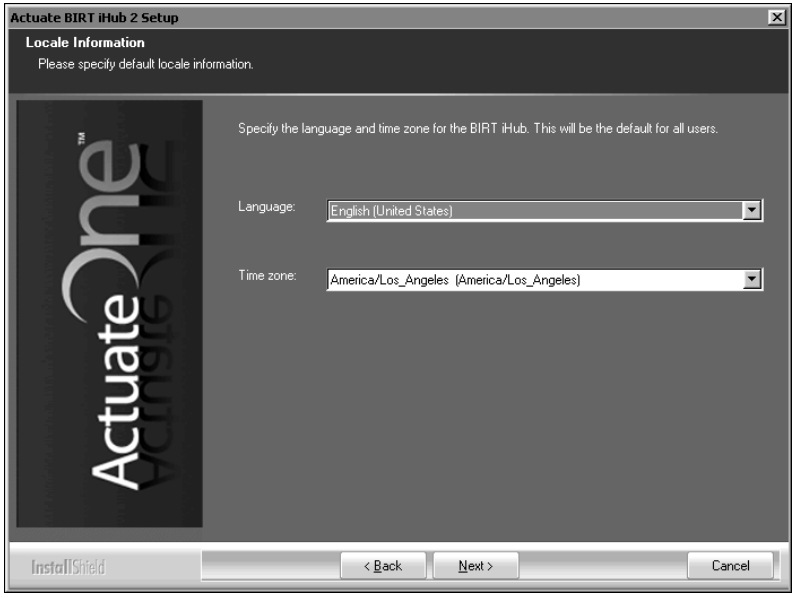


Figure 3-13 Specifying locale information

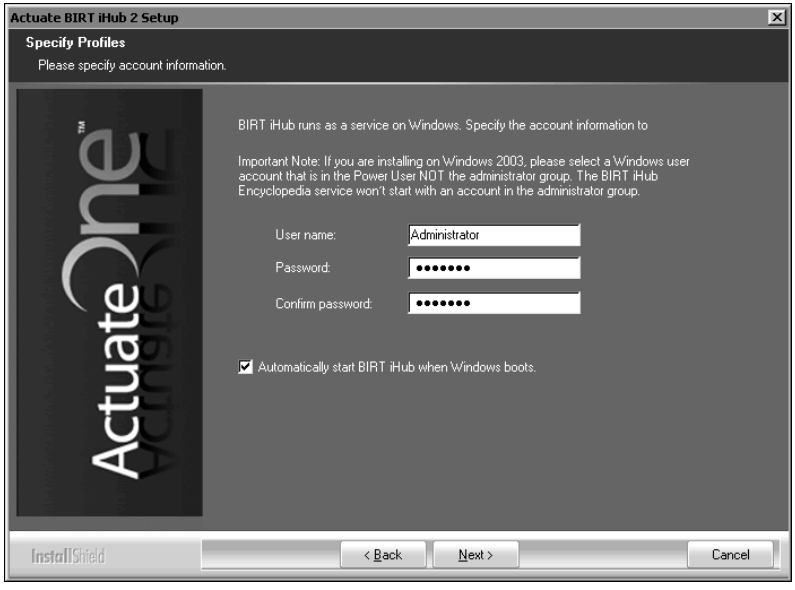


Figure 3-14 Specifying the account for running the iHub service

If prompted to add Log on as a service privilege, choose Yes.

11 In System Configuration Password, type and confirm a password for Configuration Console, as shown in Figure 3-15. For both Configuration Console and Management Console, the default user name is Administrator. The Administrator account for Management Console has no password initially. You can log in to these consoles and change the password settings after installing iHub. Choose Next.

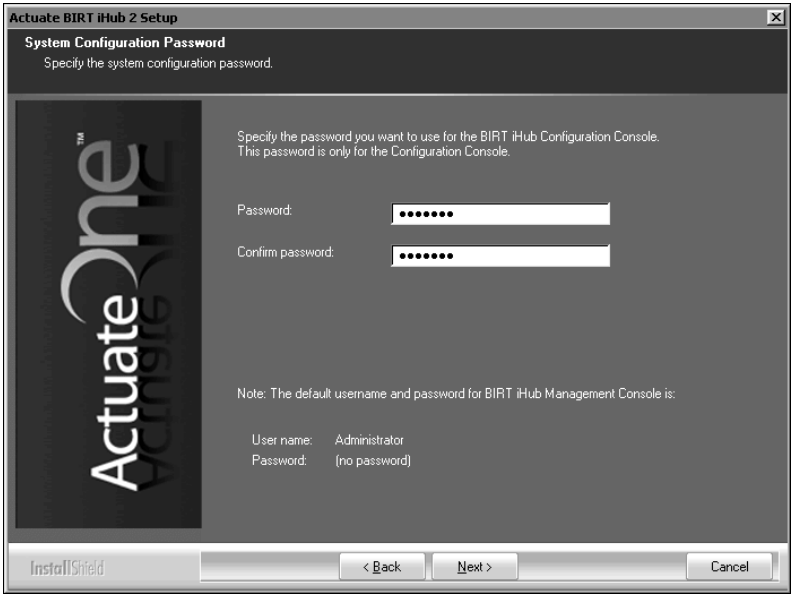


Figure 3-15 Specifying the password for using Configuration Console

12 In Start Copying Files, review the settings shown in Figure 3-16. Choose Next.

Setup Status displays an indicator showing how the installation is progressing, as shown in Figure 3-17.

Figure 3-16 Reviewing settings before copying files

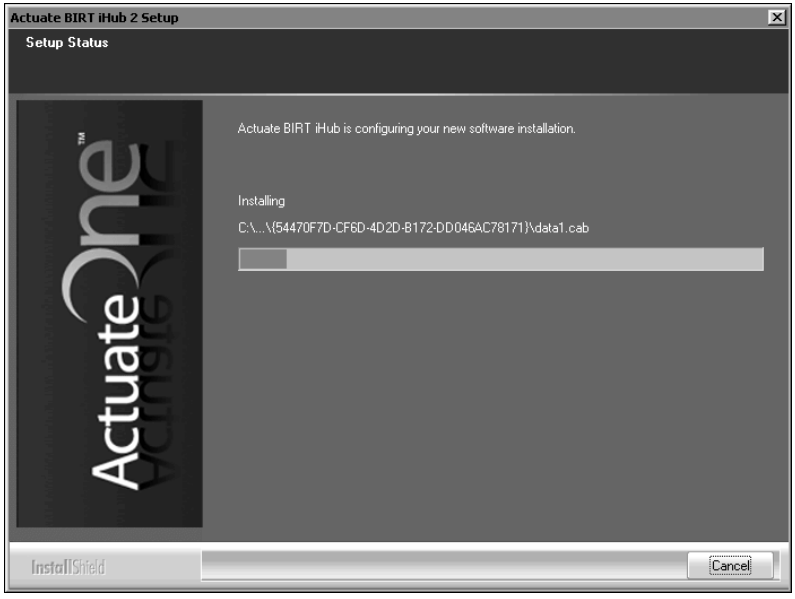


Figure 3-17 Viewing setup status

13 Choose Finish to exit the wizard, as shown in Figure 3-18.

Figure 3-18 Exiting the installation wizard

14 The installation program prompts you to install the online help from the following location, as shown in Figure 3-19:

```
http://www.actuate.com
```

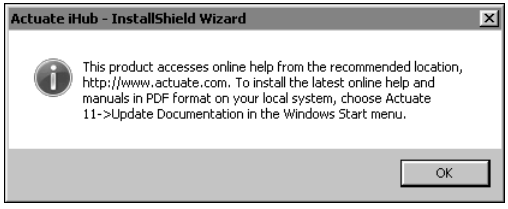


Figure 3-19 Viewing the install online help and manuals prompt

To install the online help and PDF manuals from this location, in Windows, choose Start➤ Actuate➤Update Documentation.

The installation program installs shortcuts on the desktop, as shown in Figure 3-20.

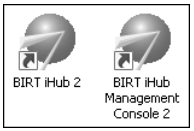


Figure 3-20 Viewing iHub shortcuts on the desktop

These shortcuts provide access to the following iHub components:

- BIRT iHub Management Console 2
  Launches Management Console to set up user accounts and run designs
- BIRT iHub 2
  Opens Welcome to Actuate iHub from which you can log in to Information Console to perform tasks, such as accessing folders and viewing documents

Chapter

4

Migrating to BIRT iHub

This chapter discusses the following topics:

- Migrating to BIRT iHub
- Preparing to migrate to BIRT iHub
- Performing a side-by-side migration

Migrating to BIRT iHub

When migrating to BIRT iHub, the administrator first installs BIRT iHub in a stand-alone, side-by-side configuration with an existing BIRT iServer Release 11 system. After installation, the administrator reconfigures BIRT iHub to use the existing BIRT iServer relational database management system (RDBMS) that contains the system and Encyclopedia volume metadata and related data stored on disk.

The administrator installs and migrates to BIRT iHub by performing the following tasks:

- Shuts down the BIRT iServer Release 11 system and, if using the embedded out-of-the-box (OOTB) PostgreSQL RDBMS, shuts down the Windows Actuate 11 PostgreSQL for BIRT iServer service. It is not necessary to shut down the service for an alternative RDBMS, such as DB2, Microsoft SQL Server, Oracle, or a pre-existing PostgreSQL system.
- Installs a stand-alone BIRT iHub System that uses the embedded OOTB PostgreSQL RDBMS, then shuts down the BIRT iHub System and PostgreSQL RDBMS.
- Backs up the existing BIRT iServer Release 11 System to safeguard against failure, including all metadata, data, and necessary configuration files.
- Performs the manual operations necessary to reconfigure BIRT iHub to use the existing BIRT iServer RDBMS that contains the system and Encyclopedia volume metadata and related data stored on disk.
- If migrating from a BIRT iServer installation earlier than Release 11 Service Pack 4, such as Release 11 Service Pack 1, 2, or 3, runs the Ant script, a11Upgrade.xml, to upgrade the Encyclopedia volume schema before restarting BIRT iHub.
- Restarts BIRT iHub System.

When migrating to BIRT iHub from an earlier iServer installation, such as Release 10 Service Pack 1, the administrator must first upgrade to Release 11, then migrate to iHub.

Preparing to migrate to BIRT iHub

Before performing a BIRT iHub migration, the system administrator must prepare the environment by configuring a dedicated user account with administrator privileges for installing and running BIRT iHub. If installing BIRT iHub side-by-side on the same machine as BIRT iServer, use the existing account reserved for BIRT iServer operations. For more information on configuring a

dedicated user account with administrator privileges for installing and running BIRT iHub, refer to Chapter 2, “Installing BIRT iHub,” earlier in this book.

The following sections describe how to migrate to BIRT iHub using the side-by-side migration process.

Performing a side-by-side migration

In the side-by-side migration process, the administrator uses the wizard-based installation program to install a stand-alone BIRT iHub system with an embedded OOTB PostgreSQL RDBMS in a path separate from an earlier iServer Release 11 installation. After installing, the administrator reconfigures the BIRT iHub system to use the RDBMS in the iServer Release 11 system.

BIRT iHub supports using the embedded OOTB PostgreSQL RDBMS or an alternative RDBMS, such as DB2, Microsoft SQL Server, Oracle, or a pre-existing PostgreSQL system. For more information on installing a BIRT iHub system that uses one of these RDBMS, see Chapter 3, “Installing BIRT iHub using an alternative database.”

Installing BIRT iHub system

The following procedure shuts down the BIRT iServer Release 11 system and installs BIRT iHub in a stand-alone, side-by-side configuration.

How to install BIRT iHub system

1. Shut down the BIRT iServer Release 11 system by performing the following tasks:
    1. Log on to Configuration Console. In the Simple view, choose Stop system. Log out of Configuration Console.
    2. Open Windows Services and perform the following tasks:
        1. Right-click Actuate 11 BIRT iServer service and choose Properties.
        2. On Actuate 11 BIRT iServer Properties—General, in Startup type, select Manual to disable automatic startup, then choose Stop to stop the service. Choose OK.
        3. If using the embedded OOTB PostgreSQL RDBMS, select Actuate 11 PostgreSQL for BIRT iServer service, then choose Stop the service to shut it down.

            It is not necessary to shut down the service for an alternative RDBMS, such as DB2, Microsoft SQL Server, Oracle, or a pre-existing PostgreSQL system.

2 Install a stand-alone BIRT iHub system that uses the embedded OOTB PostgreSQL RDBMS. Install the embedded OOTB PostgreSQL RDBMS to listen at a port different from the existing RDBMS used in the BIRT iServer Release 11 installation.

   When installing BIRT iHub, use the same system name as the previous BIRT iServer Release 11 installation. iHub inserts this name into the names it creates for the Encyclopedia volume schema and the iHub system schema. Additionally, iHub assigns this name to the default Encyclopedia volume. For information on how to install a stand-alone BIRT iHub system, refer to Chapter 2, "Installing BIRT iHub," earlier in this book.

The following procedure describes step-by-step how to perform the manual operations required to complete the migration of Release 11 BIRT iServer to BIRT iHub.

Performing a manual side-by-side migration

After installing the stand-alone version of BIRT iHub in a path separate from the earlier BIRT iServer Release 11 system, the administrator completes the migration by performing the following tasks:

- Backs up the existing BIRT iServer Release 11 system
- Shuts down the BIRT iHub system and the embedded OOTB PostgreSQL RDBMS
- Performs the manual operations necessary to reconfigure iHub to use the BIRT iServer RDBMS containing the system and Encyclopedia volume metadata and related data stored on disk

This procedure replaces the BIRT iServer Release 11 installation with a new BIRT iHub system. Do not run the iServer Release 11 program files after performing these operations.

The following section describes how to perform these operations.

How to perform a manual side-by-side migration

1 Backup the existing BIRT iServer Release 11 system to safeguard against failure, including all metadata, data, and necessary configuration files.

   For information on how to back up the iHub or iServer system and Encyclopedia volume schemas and related data and configuration files, refer to Chapter 10, "Backing up an Encyclopedia volume," later in this book.

2 Shut down the BIRT iHub system by performing the following tasks:

   1 Log on to Configuration Console. In the Simple view, choose Stop system. Log out of Configuration Console.

2. In Windows Services, stop Actuate BIRT iHub service, then stop Actuate PostgreSQL for BIRT iHub service.

3. Copy the BIRT iServer Release 11 acserverconfig.xml and, if migrating from BIRT iServer 11 Service Pack 4, the encryption keys file from the BIRT iServer configuration folder to the BIRT iHub configuration folder. For example, in a default iServer Release 11 Service Pack 4 installation, copy these files from the following location:

   ```
   C:\Actuate\iServer\data\config\11SP4
   ```

   to:

   ```
   C:\Actuate\iHub\data\config\iHub2
   ```

4. If the BIRT iServer Release 11 system uses the out-of-the-box (OOTB) PostgreSQL RDBMS, give the BIRT iHub system access to the PostgreSQL data in the BIRT iServer Release 11 system by performing the following tasks:

   1. Copy the postgresql folder from the iServer AC_DATA_HOME\encyc folder to the iHub AC_DATA_HOME\encyc folder.

   2. From the iHub AC_DATA_HOME\encyc folder, navigate to the following location:

      ```
      \postgresql\data
      ```

   3. In a text editor, open the postgresql.conf file and find the second occurrence of log_directory. In a default BIRT iServer Release 11 installation, the log_directory setting points to the following location:

      ```
      log_directory = 'C:/Actuate/iServer/data/postgresql/log'
      ```

      Edit this setting to point to the location of the iHub AC_DATA_HOME \postgresql\log folder. For example, in a default iHub installation, edit the setting to point to the following location:

      ```
      log_directory = 'C:/Actuate/iHub/data/postgresql/log'
      ```

   4. Scroll down until you see the entry for data_dictionary. In a default BIRT iServer Release 11 installation, the data_directory setting points to the following location:

      ```
      data_directory = 'C:/Actuate/iServer/data/encyc/postgresql
         /data'
      ```

      Edit this setting to point to the location of the iHub AC_DATA_HOME \encyc\postgresql\data folder. For example, in a default iHub installation, edit the setting to point to the following location:

      ```
      data_directory = 'C:/Actuate/iHub/data/encyc/postgresql
         /data'
      ```

   5. Exit the editor, saving the file to disk.

5. Update the AC_DATA_HOME entry in the BIRT iHub acpmdconfig.xml file to use the value in the BIRT iServer acpmdconfig.xml file. For example, perform the following tasks:

   1. In a BIRT iServer Release 11 Service Pack 4 installation, navigate to the following location:

      ```
      C:\Program Files (x86)\Actuate11SP4\iServer\etc
      ```

   2. In a text editor, open the acpmdconfig.xml file, and copy the following setting:

      ```
      <AC_DATA_HOME>
         C:\Actuate\iServer\data
      </AC_DATA_HOME>
      ```

   3. In the iHub installation, navigate to the following location:

      ```
      C:\Program Files (x86)\Actuate\iHub2\etc
      ```

   4. In a text editor, open the acpmdconfig.xml file, and replace the default AC_DATA_HOME setting with the value from the BIRT iServer Release 11 acpmdconfig.xml file, as shown in Listing 4-1. Exit the editor, saving the file to disk.

      Listing 4-1 acpmdconfig.xml

      ```
      <PMDConfig>
         ...
         <!--Server information -->
         <Server>
         ...
            <AC_DATA_HOME>C:\Actuate\iServer\data</AC_DATA_HOME>
         ...
      ```

6. If the BIRT iServer Release 11 system uses the out-of-the-box (OOTB) PostgreSQL RDBMS, start the Actuate PostgreSQL for BIRT iHub service.
7. If migrating from an installation earlier than Release 11 Service Pack 4, such as Release 11 Service Pack 1, 2, or 3, upgrade the Encyclopedia volume schema by performing the tasks described in "How to upgrade the Encyclopedia volume schema when migrating from an installation earlier than Release 11 Service Pack 4," later in this chapter.
8. Restart BIRT iHub system by performing the following tasks:

   1. In Windows Services, start Actuate BIRT iHub service.
   2. Log on to the BIRT iHub Configuration and Management Consoles to verify that the system is functioning properly, the Encyclopedia volume is online, and all migrated partitions and data are accessible.

How to upgrade the Encyclopedia volume schema when migrating from an installation earlier than Release 11 Service Pack 4

If migrating from an installation earlier than Release 11 Service Pack 4, such as Release 11 Service Pack 1, 2, or 3, upgrade the Encyclopedia volume schema by performing the following tasks:

1. In Windows System Properties—Environment Variables, add the paths to the Ant and Java executable programs to the Path system variable. For example, add the following notation to the Path variable:

   ```
   %ANT_HOME%\bin;%JAVA_HOME%\bin
   ```

   With the ANT_HOME system variable set to the following path:

   ```
   C:\Program Files (x86)\Actuate\iHub2\tools\apache-ant-1.8.2
   ```

   And the JAVA_HOME system variable set to the following path:

   ```
   C:\Program Files (x86)\Common Files\Actuate\22.0\JDK160
   ```

2. Verify that the following JAR files are in the BIRT iHub AC_SERVER_HOME /Jar folder:

   - AcCommon.jar
   - com.actuate.common-server.jar
   - iserver.jar
   - JDBCDrivers.jar
   - ServerAdminTools.jar

3. In the BIRT iHub installation, replace the existing a11Upgrade.xml script with the latest version from the BIRT iHub download page by performing the following tasks:

   1. Download the latest version of the a11Upgrade.xml script from the BIRT iHub download page.
   2. In the BIRT iHub installation, navigate to the location of the upgrade files:

      ```
      AC_SERVER_HOME\tools\upgrade\a11
      ```

      In a default Windows installation, the path is:

      ```
      C:\Program Files (x86)\Actuate\iHub2\tools\upgrade\a11
      ```

   3. Delete the existing a11Upgrade.xml script. Copy the latest version from the BIRT iHub download page to this location.

4. In this location, upgrade the Encyclopedia volume schema by performing the following tasks:

   1. In a text editor, open the a11Upgrade.properties file and edit these settings to point to the following locations:

- AC_SERVER_HOME to the iHub program files
- AC_DATA_HOME to the iServer data folder
- AC_CONFIG_HOME to the iHub configuration files

Use forward slashes in the Windows path specifications in this property file. For example, in a default installation, edit the settings to point the following locations:

```
AC_SERVER_HOME=C:/Program Files (x86)/Actuate/iHub2
AC_DATA_HOME=C:/Actuate/iServer/data
AC_CONFIG_HOME=C:/Actuate/iHub/data/config/iHub2
```

Exit the editor, saving the file to disk.

2 Open a command prompt, navigate to the location of the BIRT iHub upgrade files, and run the following command to execute the a11Upgrade.xml Ant script:

```
ant -f a11Upgrade.xml
```

Chapter

5

Installing a BIRT iHub cluster

This chapter discusses the following topics:

- Installing a BIRT iHub cluster node
- Performing a wizard-based cluster node installation
- Adding a node to a cluster

Installing a BIRT iHub cluster node

A node is a machine running a BIRT iHub instance. An iHub administrator adds a node to an iHub cluster to improve availability and throughput and scale the cluster installation to necessary processing requirements.

There are two methods of adding a new node to the cluster:

- Perform an automated, custom installation, using the wizard-driven installation program.
- Perform a manual installation or cloud deployment, using a prepared image of an installed iHub run-time environment.

Every cluster node must have network access to the following directory and resources to join the cluster:

- The shared configuration home directory
- Cluster resources, such as printers, database systems, and disk storage systems

Each node gets its configuration from a template in acserverconfig.xml, which is located in a shared configuration home directory along with the license file, acserverlicense.xml.

The acserverconfig.xml file contains the server templates as well as other configuration parameters specifying the host names, volume names, port numbers, printers, and services used by nodes in the cluster. When the Process Management Daemon (PMD) starts up, it reads these configurations and exposes the settings to the process environment variable list. When a node joins a cluster, it configures itself using its designated template.

After installation and configuring the appropriate environment variables in acpmdconfig.xml, the administrator launches the installed iHub image from the command line by passing the necessary arguments or creates a script to execute the command. Nodes with the same cluster ID, running on the same sub-net, automatically detect and join each other to form the cluster. This feature is known as elastic iHub clustering.

The cluster communicates across the network using standard HTTP/IP addressing. The cluster automatically detects the on-off status of any node. Single-point node failure does not affect the availability of other nodes.

One or more nodes in the cluster manage the request message routing. The Process Management Daemons (PMDs) located on each node coordinate processing among available iHub services based on message type to balance load across the nodes.

iHub instances running on multiple machines maintain iHub system and Encyclopedia volume metadata in databases and control access to shared volume data. The volume data can be on machines that are not running iHub, but must be shared and accessible to each iHub instance.

This loosely coupled cluster model provides the following maintenance and performance benefits:

- Startup and shutdown of an iHub is fast because it is independent of the RDBMS that manages the Encyclopedia volume. An RDBMS can remain online when shutting down an iHub and the RDBMS is available when the iHub starts up.
- Controlling the sequence of Encyclopedia volume startup is not necessary. All volumes are either already online in the RDBMS or come online as the RDBMS starts.
- Downtime to apply a patch fix patch or a diagnostic fix for an iHub is reduced. The RDBMS, including the OOTB PostgreSQL database server, does not have to be shutdown. In an iHub cluster, the patch or diagnostic fix can be applied to one iHub node at a time.

This operational model lends itself well to grid, cloud, and other data-center types of deployments. For more information about the pre-packaged Actuate cloud computing deployment option, see Chapter 6, "Installing BIRT iHub in a cloud," later in this book. For more information about administering an installed iHub cluster, see Chapter 9, "Clustering," in *Configuring BIRT iHub.*

Preparing the iHub cluster environment

This section assumes the administrator has already performed a typical, wizard-based installation to create a stand-alone iHub instance on one computer. This machine contains the shared configuration directory, which all nodes in the cluster access. This section refers to the machine containing the shared configuration directory as node1.

In the section, "Performing a wizard-based cluster node installation," later in this chapter, the administrator chooses the custom installation option in the wizard-based install program to create a cluster node. This cluster node, referred to as node2, accesses the shared resources on node1.

Before performing a cluster node installation, the administrator performs the following tasks:

- On node1:
  - Shares the folders that all cluster nodes access

- Turns off the Windows firewall
- Obtains the machine host name and IP address
- Tests the network accessibility of the machine
- Reconfigures the partition path for the default partition in Configuration Console using UNC format
- Edits acserverconfig.xml and sets a property to make the metadata database visible to all nodes

- On node2:
  - Turns off the Windows firewall
  - Obtains the machine host name and IP address
  - Tests the network accessibility of the machine

The following instructions provide a basic example of the operations required to configure network sharing and firewall settings in the Windows environment. It is the responsibility of the administrator performing the installation to make sure that all settings conform to the security policies in force for the environment.

How to share the encyc and config\iHub2 folders

In an iHub installation, cluster nodes must have read-write access to AC_DATA_HOME\encyc and AC_DATA_HOME\config\iHub2 folders on node1. To give a cluster node read-write access to the folders and files in the encyc folder and to the files in the config\iHub2 folder, perform the following tasks on node1:

1. Using Windows Explorer on node1, right-click the folder, AC_DATA_HOME \encyc. Choose Properties, as shown in Figure 5-1.
2. On encyc Properties, choose Sharing, as shown in Figure 5-2.

   On Sharing, choose Advanced Sharing.

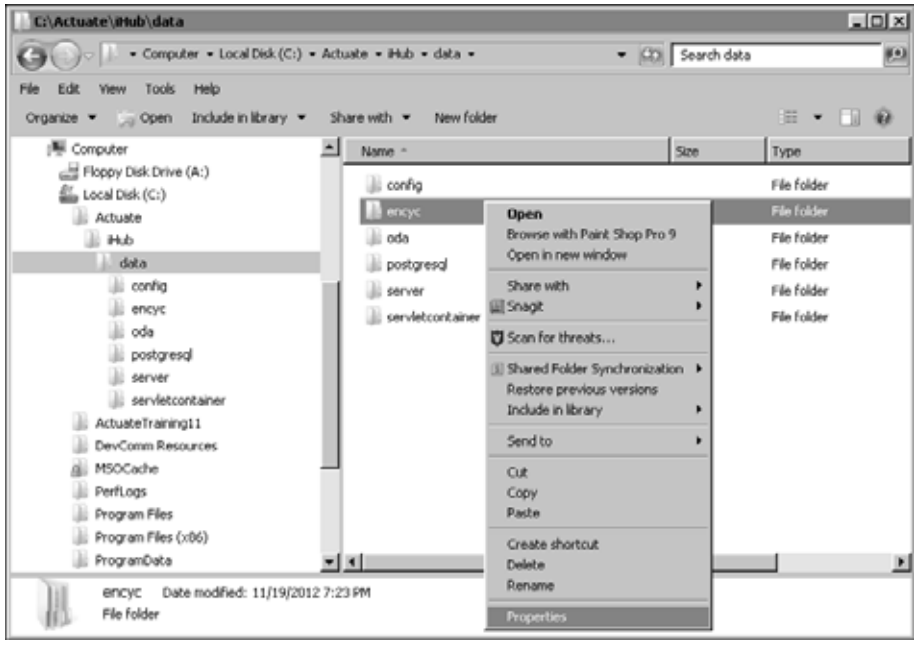


Figure 5-1 Choosing Sharing and Security

Figure 5-2 Sharing the Encyclopedia volume

3 On Advanced Sharing, select Share this folder, as shown in Figure 5-3

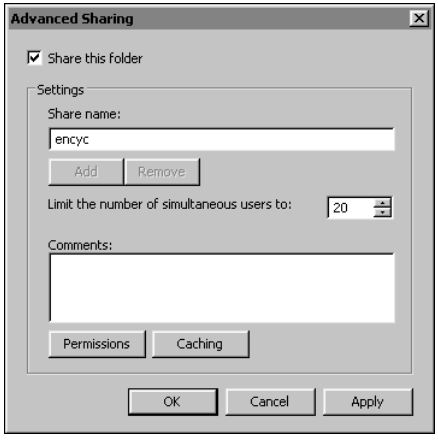


Figure 5-3 Selecting Share this folder

On Advanced Sharing, choose Permissions.

4 On Permissions for encyc, in Share Permissions, select Allow for Change and Read, as shown in Figure 5-4.

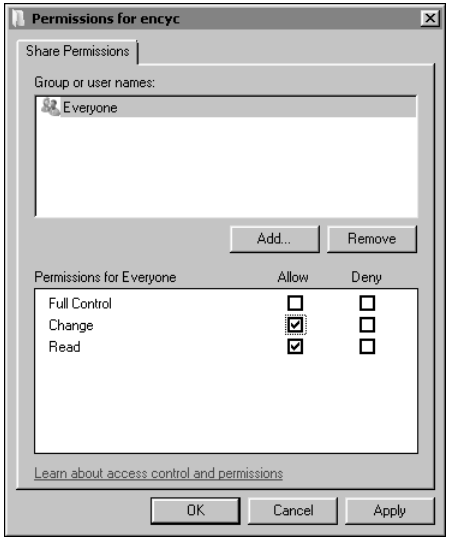


Figure 5-4 Selecting Change and Read permission

Choose OK.

On Advanced Sharing, choose OK.

On encyc Properties, choose Close.

5 Repeat steps 1 through 4 for the \config\iHub2 folder. Make sure that all settings conform to the security policies in force for the environment.

Close Windows Explorer.

How to turn off the Windows firewall

Perform the following steps on node1 and node2:

1 Choose Start➤Control Panel➤System and Security➤Windows Firewall.

2 On Windows Firewall, choose Turn Windows Firewall on or off. Make sure that all settings conform to the security policies in force for the environment.

3 On Customize Settings, in Home or work (private) network location settings, choose Turn off Windows Firewall, as shown in Figure 5-5.

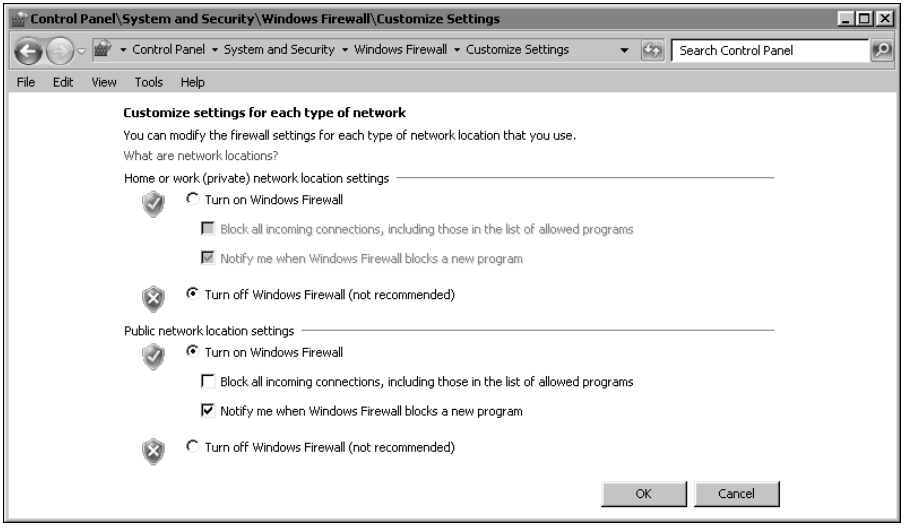


Figure 5-5 Turning off the home or work network location firewall

Close Windows Firewall.

How to display a computer's IP address

To obtain the host names of node1 and the computer on which you will install the cluster node, perform the following tasks on node1 and node2:

1 Choose Start➤Programs➤Accessories➤Command Prompt.

2 In Command Prompt, type the following command:

```
ipconfig /all
```

Press Enter. The host name appears, as shown in Figure 5-6. In this example, the host name for node1 is urup.

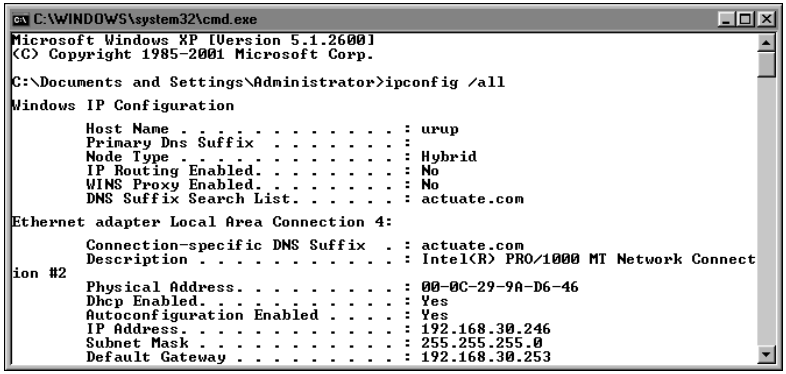


Figure 5-6 Displaying the host name

3 Write the host names and IP addresses of the computers to be clustered, as shown in Table 5-1.

Table 5-1 Host names and IP addresses of computers to be clustered

iHub	Host name	IP address
Node1	urup	192.168.30.246
Node2	kozu	192.168.30.233

How to test the connection between computers

Perform the following steps on both computers:

1 In Command Prompt, type the ping command followed by the IP address or host name of the other computer. For example, type the following command to ping a computer named kozu:

```
ping kozu
```

Press Enter.

If your computer reaches the other computer, Command Prompt displays a series of replies, as shown in Figure 5-7.

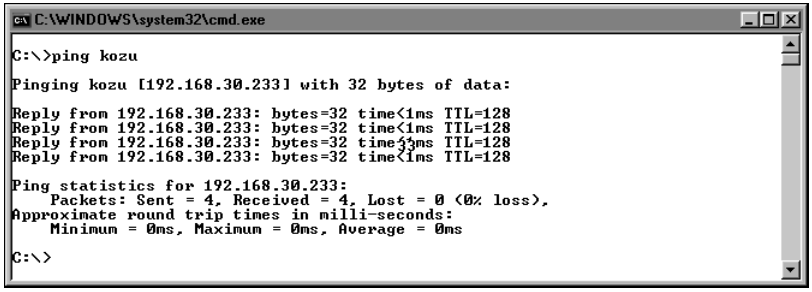


Figure 5-7 Receiving a reply to a ping command

2 Close Command Prompt.

How to prepare node1 for clustering

On node1, or urup in this example, perform the following tasks:

1 Log in to Configuration Console and choose Advanced view. From the side menu, choose Volumes. On Volumes, point to the icon next to the default volume, corp in this example, and choose Take offline, as shown in Figure 5-8.

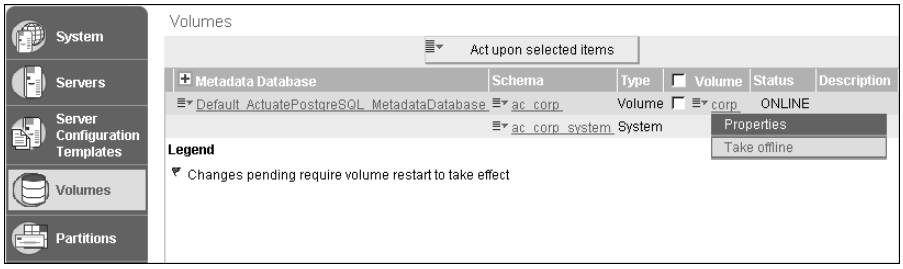


Figure 5-8 Taking a volume offline

In Volume Offline Grace Period, choose OK.

From the side menu, choose Partitions.

2 On Partitions, point to the icon next to DefaultPartition and choose Template settings, as shown in Figure 5-9.

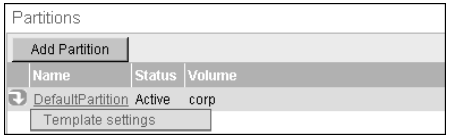


Figure 5-9 Choosing Template settings

On Template Settings, choose Change. In Partition Path, type the path to the encyc folder, using UNC format. For example, if the hostname of node1 is urup, type:

```
\\urup\encyc
```

Template Settings appears as shown in Figure 5-10.

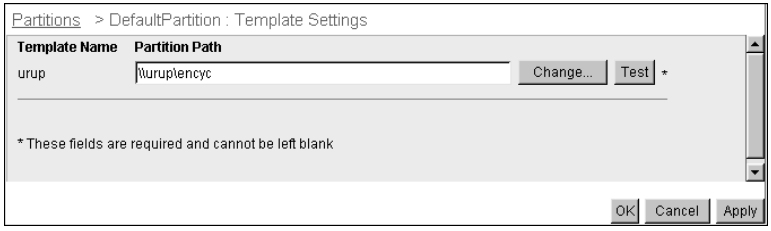


Figure 5-10 Typing the partition path

To verify that iHub can access the encyc folder, choose Test.

A message appears, stating that the test was successful, as shown in Figure 5-11. Choose OK.

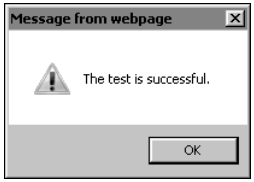


Figure 5-11 Verifying a successful partition path test

On Template Settings, choose OK.

From the side menu, choose System.

3 On System—Status, choose Stop to stop the system, as shown in Figure 5-12.

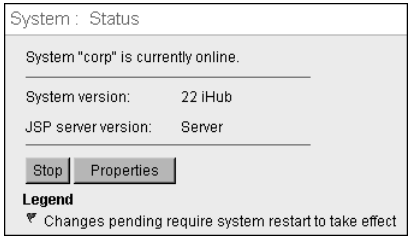


Figure 5-12 Choosing to stop the system on System Status

4 Log out of Configuration Console.

5 Choose Start➤Settings➤Control Panel.

In Control Panel, choose Administrative Tools—Services.

On Services, stop the Actuate BIRT iHub service.

6 Open Windows Explorer, and navigate to AC_DATA_HOME/config/iHub2. Make a backup copy of acserverconfig.xml, then open acserverconfig.xml and perform the following tasks:

1 Locate the <ConnectionProperties> element under the <MetadataDatabase> element.

2 Under <ConnectionProperties>, locate:

```
<ConnectionProperty
      Name="server"
      Value="localhost"/>
```

3 Change Value from localhost to the name of the machine on which the volume resides, in this example "urup". For urup, the volume is located on localhost, but in a cluster setting the administrator has to specify the machine name, since the volume is not on a localhost from the reference point of the other nodes.

4 Save acserverconfig.xml, and exit the file.

7 On Services, start the Actuate BIRT iHub service.

Creating an account with Windows administrator privileges

Before installing iHub, create a Windows user account that is a member of the Administrators group. Use this account when installing and running iHub.

The iHub user account must meet the following requirements:

- Be a member of the Windows Administrators group.
  The account must have privileges to access the required software and hardware, such as database servers, printers, and iHub files and folders.
- Have log on as a service privilege.
  If the account does not meet this requirement, the iHub installation program prompts you to configure the privilege to run the Actuate BIRT iHub service.

On a new Windows Vista installation, the initial user account is not a member of the Administrators group. You must configure this user account to be a member of this group.

When installing iHub in Windows 2003, create a Windows user account that is a member of the Power Users not the Administrators group. Make sure that the Account has permission to access any printers required for printing. Perform the installation using an account that has Administrator privileges. During the installation, when prompted to specify the user account to run the iHub service, specify the Power User account.

For more information about configuring a Power User and iHub account and log on as a service privilege, see "Creating an account with Windows administrator privileges" in Chapter 2, "Installing BIRT iHub."

Performing a wizard-based cluster node installation

This section assumes the Administrator has already performed the wizard-based Typical install to create a stand-alone iHub on one computer and has performed the tasks described in the Preparing the iHub cluster environment section, earlier in this chapter. To create a stand-alone iHub installation, see "Performing a new installation" in Chapter 2, "Installing BIRT iHub." To prepare the

stand-alone iHub node for clustering, see “Preparing the iHub cluster environment,” earlier in this chapter.

When creating a BIRT iHub cluster, the Administrator must install and run all cluster nodes using the same administrative user account. The following section describes how to install an iHub cluster node in the Windows operating system using the installation wizard.

How to install a cluster node in Windows

1. Download the iHub distribution package from an FTP software distribution site. Extract the files. Run the self-extracting executable file, ActuateBIRTiHub.exe. The welcome message appears, as shown in Figure 5-13. Choose Next.

Figure 5-13 Viewing the welcome message

2. Read and accept the license agreement, as shown in Figure 5-14. Choose Next.
3. The setup installs the prerequisite components that BIRT iHub requires, as shown in Figure 5-15. Choose Next.

Figure 5-14 Accepting the license agreement

Figure 5-15 Installing Prerequisites

4 In Setup Type, select Custom, as shown in Figure 5-16.

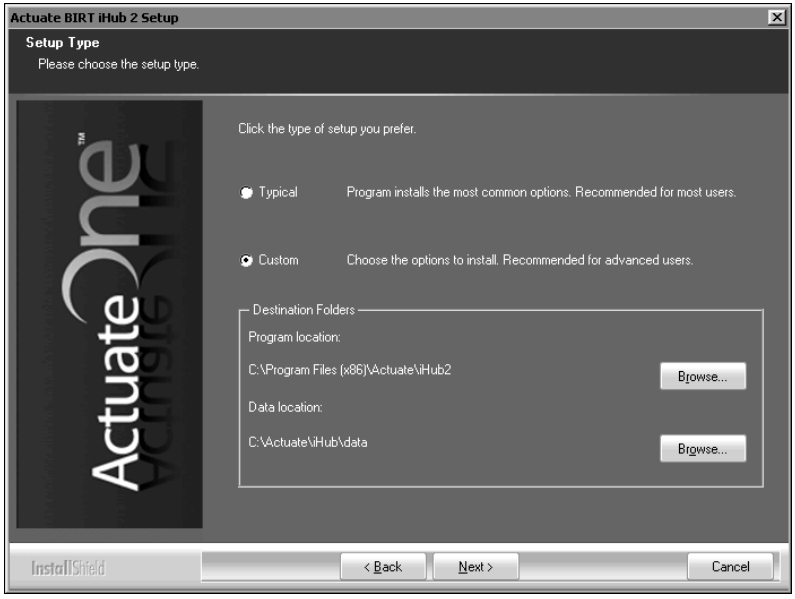


Figure 5-16 Selecting the custom installation

In Destination Folder, accept the default or choose a new destination folder for the Program and Data locations. Choose Next.

iHub uses the Program location to resolve the paths to all the binaries that it launches. The environment variable, AC_SERVER_HOME, points to the location of the iHub binaries. The default path for the program location is C:/Program Files (x86)/Actuate/iHub2.

iHub uses the Data location to store the iHub logs, iHub encyclopedia, including the PostgreSQL data, and all other run-time data. The environment variable, AC_DATA_HOME, points to the location of the iHub data. The default path for the data location is C:/Actuate/iHub/data.

Each iHub instance must to have its own AC_SERVER_HOME and AC_DATA_HOME folders. These folders cannot be shared by other nodes in a cluster.

5 In Select Features, select all features—Core products, Management Console, and Examples, as shown in Figure 5-17. A cluster node must have access to Configuration Console. Configuration Console installs with Management Console. Choose Next.

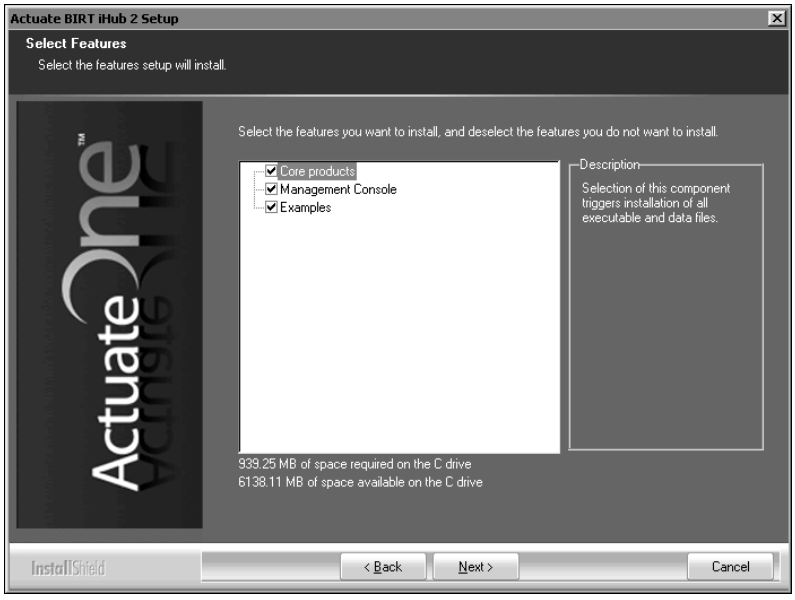


Figure 5-17 Selecting features to install

6 In iHub Installation Option, select Cluster node to install iHub as a cluster node. Then specify the configuration home location, as shown in Figure 5-18.

Figure 5-18 Choosing the option to install a node

The configuration home location is the shared directory containing the acserverconfig.xml, acserverlicense.xml, and acconfigowner.lock files for the cluster. The administrator should specify the location using the Universal Naming Convention (UNC) Format.

In an iHub installation, the configuration files are located in AC_DATA_HOME\config\iHub2 by default. If the shared configuration folder is in this default location on a server named urup, then the administrator should specify \\urup\iHub2 as the path.

Choose Next.

7 In License File Details, select Use the license that you purchased. Choose Browse then navigate to and choose the license file, as shown in Figure 5-19. Choose Next. Alternatively, choose Try out the product using the included evaluation software license if you do not have a purchased license.

Figure 5-19 Specifying the license file

If installing using a named user license, a prompt appears advising you to check that the volume does not exceed the number of registered users authorized by the license, as shown in Figure 5-20.

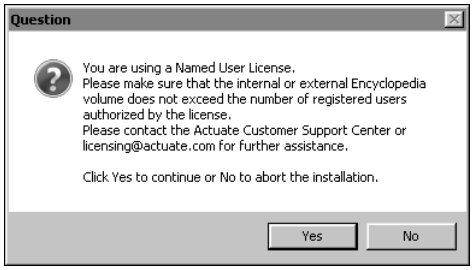


Figure 5-20 Viewing the named user license question

Choose Yes to continue the installation.

8 In Locale Information, choose Next to accept the default language and time zone, as shown in Figure 5-21. Alternatively, choose the language and locale settings for your region.

Figure 5-21 Specifying a language

9 In Specify Profiles, type the user name, password, and confirm the password for the account used to start the Actuate BIRT iHub service, as shown in Figure 5-22. The account must be a member of the Administrators group. Actuate recommends that you limit access to this account for security reasons.

If you are installing on Windows 2003, specify a user account that is in the Power Users not the Administrators group. A user account in the Administrators group cannot start the Actuate BIRT iHub service.

Accept Automatically start iHub when Windows boots, as shown in Figure 5-22. If you deselect this option, you must start the service manually from Windows Services. Choose Next.

If prompted to add Log on as a service privilege, choose Yes.

Figure 5-22 Specifying the account for running the iHub service

10 In iHub Configuration, shown in Figure 5-23, type the cluster node name or IP address for the node. Type a port number or accept the default port number, 8100. The Actuate BIRT iHub service on Windows binds to this port number to listen for requests. Choose Next.

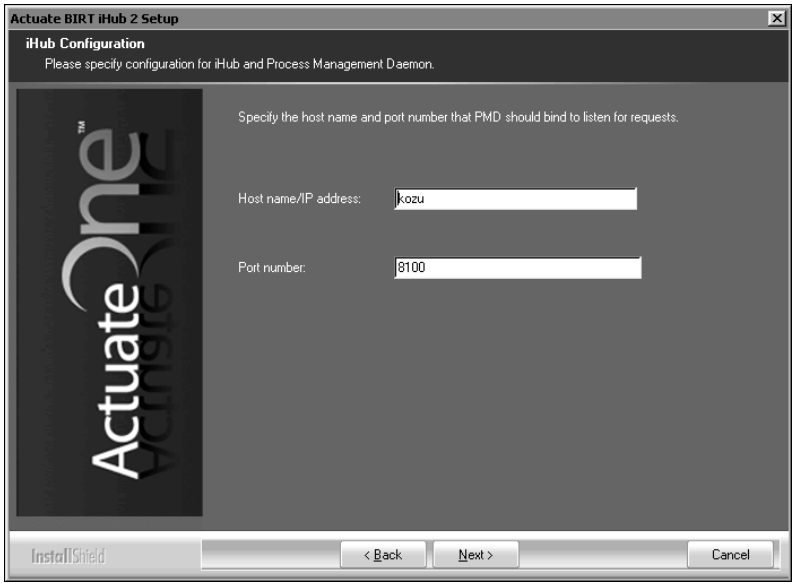


Figure 5-23 Identifying the cluster node and port to bind iHub service

11 In System Configuration Password, type a Configuration Console password, as shown in Figure 5-24. Note that the default user name for Management Console is Administrator with no password. Choose Next.

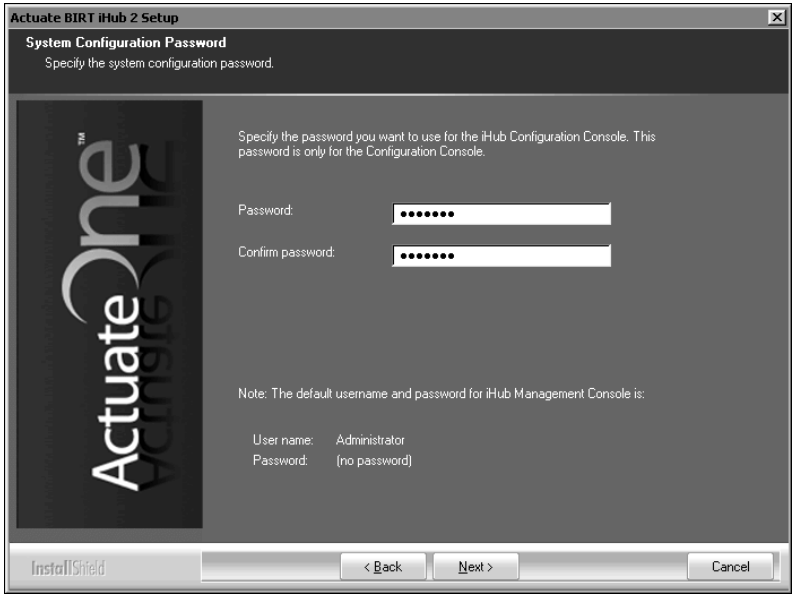


Figure 5-24 Specifying the password for using Configuration Console

12 If you chose to install Management Console in step 5, complete the following steps:

1. Specify the following information, as shown in Figure 5-25. Choose Next:
   - Host name and port number for the following items:
     - PMD (Process Management Daemon) Configuration
     - iHub Configuration
   - Default volume name

     The BIRT iHub System name the Administrator specified on Encyclopedia Metadata Storage and System Name during the wizard-based Typical install to create a stand-alone iHub, as shown in Figure 2-5 in "Performing a new installation" in Chapter 2, "Installing BIRT iHub."

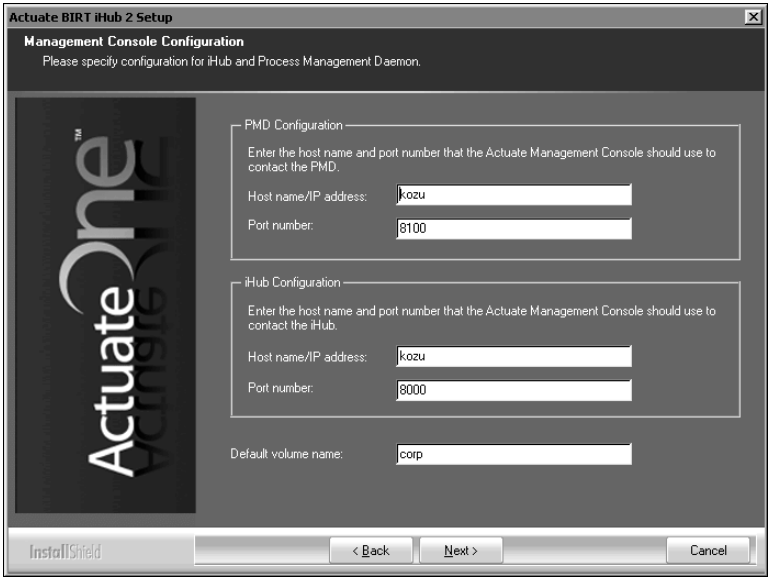


Figure 5-25 Specifying the Management Console Configuration

2. In Actuate iHub Application Container, type the port number that the iHub Application Container uses, or accept the default port number, 8900, as shown in Figure 5-26. Choose Next.

Figure 5-26 Specify the port number for iHub Application Container

3 In Context Path, type the context path for Management Console or accept the default path, /acadmin, as shown in Figure 5-27. Choose Next.

Figure 5-27 Specifying the context path

4 In Select Program Folder, specify a folder name containing the iHub icons that are used to launch the consoles, or accept the default name, Actuate, as shown in Figure 5-28. Choose Next.

Figure 5-28 Specifying a program folder

13 In Start Copying Files, review the settings shown in Figure 5-29. Choose Next to start copying files.

Figure 5-29 Reviewing settings before copying files

Setup Status displays an indicator showing how the installation is progressing, as shown in Figure 5-30.

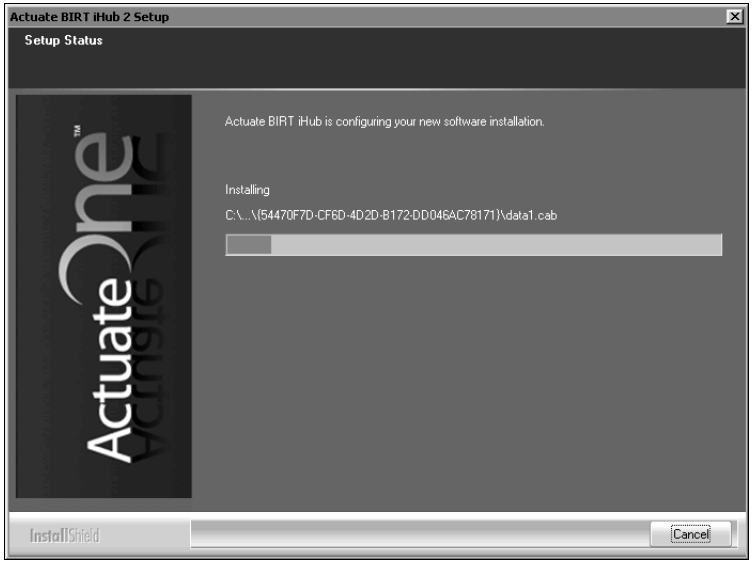


Figure 5-30 Viewing setup status

14 When the setup success message appears, as shown in Figure 5-31. Choose Finish to exit the wizard.

Figure 5-31 Exiting the installation wizard

15 The installation program prompts you to install the online help from the following location, as shown in Figure 5-32:

```
http://www.actuate.com
```

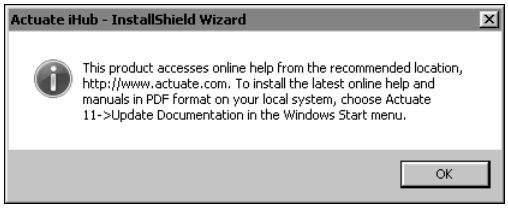


Figure 5-32 Viewing the install online help and manuals prompt

To install the online help and PDF manuals from this location, in Windows choose Start➛ Actuate➛Update Documentation.

16 If you chose to install Management Console in step 4, the installation program installs shortcuts on the desktop, as shown in Figure 5-33.

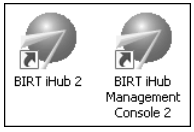


Figure 5-33 Viewing BIRT iHub shortcuts on the desktop

These shortcuts provide access to the following iHub components:

- BIRT iHub Management Console 2
  Launches Management Console to administer an Encyclopedia volume, set up user accounts, and run designs.
- BIRT iHub 2
  Opens Welcome to Actuate BIRT iHub from which you can log in to Information Console to perform tasks, such as accessing folders and viewing documents.

Adding a node to a cluster

This section assumes the administrator has completed the following tasks:

- Installed iHub as a stand-alone instance on one machine
- Prepared the node machines for clustering
- Installed iHub as a cluster node instance on a second machine

To add a node to a cluster, the Administrator performs the following tasks:

- Edits acpmdconfig.xml on the cluster node to point to the template the cluster node uses
- Starts the iHub system on the cluster node

How to add a newly installed node to a cluster

To add the newly installed node to a cluster, perform the following tasks on the cluster node:

1. Choose Start➛Settings➛Control Panel.

   In Control Panel, choose Administrative Tools—Services.

   On Services, stop the Actuate BIRT iHub service.

2. Open acpmdconfig.xml, by default located in AC_SERVER_HOME/etc, and perform the following tasks:

   1. Modify <AC_TEMPLATE_NAME> to use the name of the template that you want the node to use. The shared acserverconfig.xml contains one template by default. Specify the name that appears in the Name attribute of the <Template> element in the shared acserverconfig.xml. In this example the server template name is urup, as shown in Listing 5-1.

      Listing 5-1 Viewing the shared acserverconfig.xml

      ```
      <Templates>
         <Template
             Name="urup"
             PMDPort="8100"
             ActuateBuild="220A130126"
             ActuateVersion="2”
             ServerSOAPPort="11100"
             AppContainerPort="8900"
             RequesterRSAPIVolume="corp">
            ...
         </Template>
      </Templates>
      ```

   2. Verify that <AC_CONFIG_HOME> points to the shared configuration home directory for the cluster. This is the path you specified for the configuration home location during the install procedure, as shown in Figure 5-18.

      acpmdconfig.xml appears as shown in Listing 5-2.

      Listing 5-2 Modifying acpmdconfig.xml

      ```
      <!--Server information -->
      <Server>
         <Startup>Manual</Startup>
      ```

```
    <AC_TEMPLATE_NAME>urup</AC_TEMPLATE_NAME>
    <AC_DATA_HOME>C:\Actuate\iHub\data</AC_DATA_HOME>
    <AC_CONFIG_HOME>\\urup\iHub2</AC_CONFIG_HOME>
    ...
</Server>
<!-- Servlet Container information -->
```

3 Save acpmdconfig.xml and exit the file.

3 Start the Actuate BIRT iHub service on the cluster node.

4 On the cluster node, log in to Configuration Console, and start the system, as shown in Figure 5-34. The new cluster node will automatically contact the acserverconfig.xml in the shared configuration home directory to access its template, and then join the cluster.

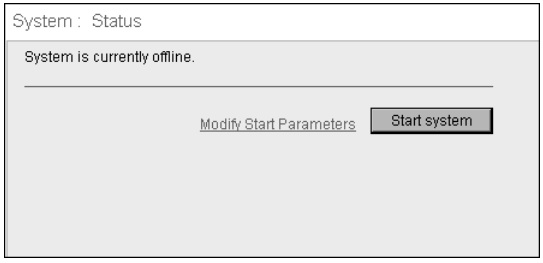


Figure 5-34 Starting the server

5 In the advanced view, choose Servers and confirm that the servers are online, and that the Factory and View services are enabled, as shown in Figure 5-35.

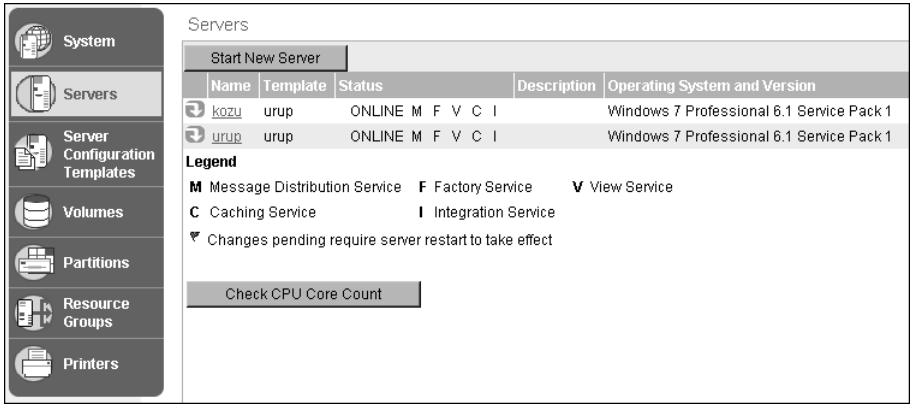


Figure 5-35 Viewing a cluster composed of urup and kozu

Chapter

6

Installing BIRT iHub in a cloud

This chapter contains the following topics:

- Understanding a BIRT iHub cloud deployment
- Deploying BIRT iHub in a cloud environment
- Setting up iHub to use the out-of-the-box (OOTB) PostgreSQL database
- Setting up iHub to use an alternative database
- Setting up iHub only
- Setting up iHub to join an existing cluster
- Understanding the cloud deployment environment

Understanding a BIRT iHub cloud deployment

A cloud deployment in the BIRT iHub environment has the following features:

- Easy-to-prepare, stateless iHub image

  This image of an installed iHub run-time environment does not require modification during installation or the life time of the instance. The administrator can create a customized image by generating an archive of an installed iHub run-time environment. Alternatively, an out-of-the-box (OOTB) image is available as a separate iHub distribution package for Windows and Linux. The administrator deploys the image by unbundling the archive and running an installation script or installing a virtual image on the target machine.

- Ready-to-launch iHub instance

  Launching an iHub instance requires minimal work. After installing, the administrator launches the deployed iHub image from the command line using a script to execute the commands.

- Elastic iHub clustering

  The use of a ready-to-launch iHub image simplifies iHub cluster installation and management. Nodes with the same cluster ID, running on the same sub-net, automatically detect and join to form the cluster. The cluster automatically detects the on-off status of any node. Single-point node failure does not affect the availability of the other nodes.

  In Release 11 and earlier, iServer used multicasting to broadcast event information and synchronize operations in a cluster. Some cloud computing environments do not support multicasting. In Release 11 Service Pack 1 and later, iServer and iHub use a third-party RDBMS as a shared repository for storing cluster information. This enhancement replaces multicasting as a way of managing the cluster environment.

  For more information on setting up a cluster after performing a cloud deployment of a stand-alone iHub installation, see "Setting up iHub to join an existing cluster," later in this chapter; Chapter 5, "Installing a BIRT iHub cluster," earlier in this book; and Chapter 9, "Clustering," in *Configuring BIRT iHub*. For more information on iHub architecture, see "Understanding Actuate BIRT iHub architecture," earlier in this book.

Deploying BIRT iHub in a cloud environment

In an Actuate cloud deployment, a stateless iHub image contains only the run-time environment. The administrator typically transfers the image of the

iHub run-time environment using a compressed archive, such as a ZIP file, or virtual image, unbundling the image on the target machine.

In addition, the administrator must install a supported version of the JAVA SE Development Kit (JDK) 1.6 (32-bit) or earlier. If not already installed on the machine, the JDK can be downloaded from the following location:

```
http://www.oracle.com/technetwork/java/javase/downloads/index.html
```

In Windows, the deployment script automatically installs the following prerequisite Microsoft Visual C++ Libraries. These libraries ship with the iHub distribution package in AC_SERVER_HOME/prerequisites. You can also download these resources from Microsoft, and manually install them.

In a 32-bit environment:

- Microsoft Visual C++ 2005 SP1 Redistributable Package (x86) - vcredist_vs2005_x86.exe
- Microsoft Visual C++ 2008 Redistributable Packages (x86) - vcredist_x86.exe

In a 64-bit environment:

- Microsoft Visual C++ 2005 Redistributable Package (x64) - vcredist_x64.exe.

If you have an earlier version of BIRT iServer installed on your machine, such as Release 10 Service Pack 1, you can continue to run the earlier version, but not simultaneously with the new version if the earlier version uses the same default ports. You must shut down the earlier version during the deployment process. The earlier version must remain shut down when the newly installed iHub is running. Reconfigure the ports for one of the versions to run both versions at the same time.

In an iHub configuration, Actuate recommends storing iHub data in a directory located outside the iHub run-time environment. In a default Windows setup performed using the wizard-based install program, the iHub run-time environment installs in the following directory:

```
C:\Program Files (x86)\Actuate\iHub2
```

The data installs in the following directory:

```
C:\Actuate\iHub\data
```

In a cloud deployment, Actuate recommends installing in an alternative directory. For example, in Windows, extract the iHub distribution package files to the following directory:

```
C:\Actuate
```

Extracting the iHub files installs the run-time environment in the following directory:

```
C:\Actuate\iHub
```

Running the setup script installs the data in the following directory:

```
C:\Actuate\iHub\data
```

The environment variable AC_SERVER_HOME points to the directory containing the run-time environment. The environment variable AC_DATA_HOME points to the directory containing the iHub data.

Deploying an iHub distribution package

In deploying an iHub distribution package, the administrator performs the following tasks:

- Extracts the contents of the iHub distribution package
- Installs a supported Java Development Kit (JDK)
- Runs the iHub setup script, installing iHub using an evaluation license

How to extract the contents of the iHub distribution package

To extract the iHub run-time resources and configure the setup script, perform the following tasks:

1. Create a new folder in a location outside of C:\Program Files or C:\Program Files (x86), such as C:\Actuate.
2. Using WinZip or another file extraction tool, extract the contents of ActuateBIRTiHub.zip to the folder created in the previous step, as shown in Figure 6-1.

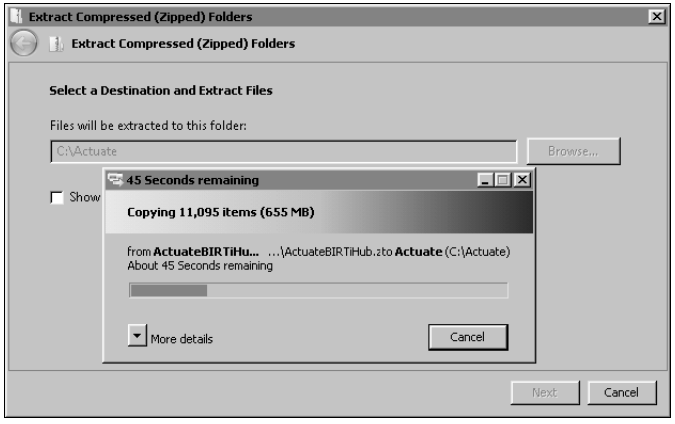


Figure 6-1 Extracting iHub distribution package contents

3. In the iHub configuration, the setting for the environment variable, AC_SERVER_HOME, specifies the location of the iHub program files. The variable, AC_JAVA_HOME, specifies the location of the Java Development Kit (JDK.)

The setupiHub.bat script automatically detects the location of AC_SERVER_HOME and AC_JAVA_HOME. If the script is unable to locate these resources in the environment, the script provides an error message. In this case, you can edit the script to set the values for these properties manually.

To specify the location of AC_SERVER_HOME and AC_JAVA_HOME in the setupiHub.bat script, perform the following tasks:

1. Using a text editor, open the script, setupiHub.bat, located in the folder where you extracted BIRT iHub.
2. Specify the paths to AC_SERVER_HOME and AC_JAVA_HOME, as shown in Listing 6-1.
3. Save and close the file.

Listing 6-1 setupiHub.bat script

```
@echo off
REM Script that sets up the iHub and starts the iHub

cls
Title Actuate BIRT iHub 2 setup and start script for
   evaluation
...
set AC_SERVER_HOME=C:\Actuate\iHub
set AC_JAVA_HOME=C:\JDK160
...
```

Running the setup script

The setup script provides the following stand-alone and cluster installation options:

- Stand-alone
  - Set up iHub to use the out-of-the-box (OOTB) PostgreSQL database.
    Sets up iHub and an embedded out-of-the-box (OOTB) PostgreSQL database for storing iHub system and Encyclopedia volume metadata.
  - Set up iHub to use an alternative database.
    Sets up iHub and an external database, such as DB2, Microsoft SQL Server, Oracle, or a pre-existing PostgreSQL database, for storing iHub system and Encyclopedia volume metadata. Requires superuser or database system administrator access to the external database.
  - Set up iHub only.
    Sets up only the iHub program files. This option requires logging into Configuration Console after installing iHub to set up a connection to the database used for storing iHub system and Encyclopedia volume metadata. Requires the database administrator to run SQL Data Definition

Language (DDL) scripts to create a database and the following schema owner and application user accounts with appropriate privileges:

- iHub system schema owner
- Encyclopedia volume schema owner
- iserver application user

For more information about creating the iHub system and Encyclopedia volume schemas and iserver user in an alternative database, see Chapter 3, "Installing BIRT iHub using an alternative database."

- Cluster
  - Set up iHub to join an existing cluster.
    Sets up an iHub node on an existing cluster to add more capacity. The setup prompts the administrator for the location of the shared configuration files.

How to run the setup script

1. From the Windows menu, choose Start➤Run.
2. Open a command prompt by typing cmd, and choose OK.
3. Navigate to the folder where you extracted the iHub package, such as C:\Actuate.
4. Type setupiHub, and press Enter to execute the installation script.
5. The script prompts you to choose one of the following stand-alone or cluster options in setting up iHub, as shown in Figure 6-2:
   - Standalone:
     - 1. Set up iHub to use the out-of-the-box (OOTB) PostgreSQL database.
     - 2. Set up iHub to use an alternative database.
       Follow instructions under iHub\tools\install\readme before selecting this choice.
     - 3. Set up iHub only.
       Use this option to setup iHub only and add an external database connection later using Configuration Console.
   - Cluster:
     - 4. Set up iHub to join an existing cluster.
     - 5. Abort to terminate the operation.

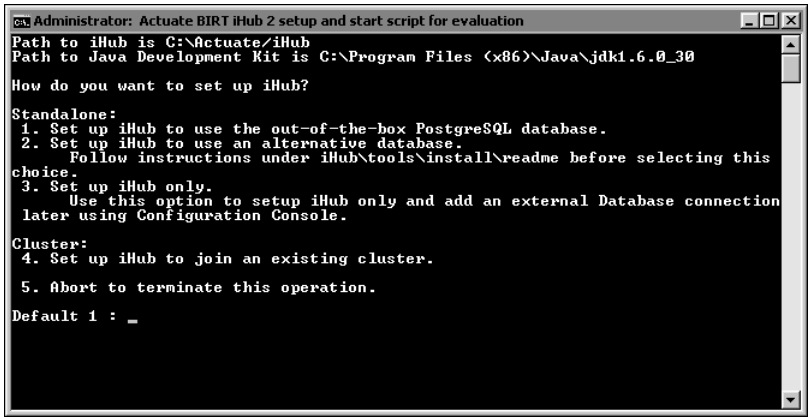


Figure 6-2 Choosing setup type

The following sections describe how to perform an installation for each option.

Setting up iHub to use the out-of-the-box (OOTB) PostgreSQL database

The following section describes how to set up iHub to use the out-of-the-box (OOTB) PostgreSQL database.

How to set up iHub to use the out-of-the-box (OOTB) PostgreSQL database

After extracting the contents of ActuateBIRTiHub.zip and running the setup script as described in "How to extract the contents of the iHub distribution package" and "How to run the setup script" earlier in this chapter, perform the following tasks:

1. In the setupiHub menu, press Enter to choose default option 1, Set up iHub to use the out-of-the-box (OOTB) PostgreSQL database, as shown in Figure 6-3.

   The script performs the following tasks, as shown in Figure 6-3:

   - Installs the prerequisite C++ run-time components
   - Sets up the iHub deployment files, including log and security keys files
   - Installs and starts the OOTB PostgreSQL database system used to store Encyclopedia volume metadata
   - Starts BIRT iHub
   - Creates the Encyclopedia volume
   - Uploads the Encyclopedia volume sample content

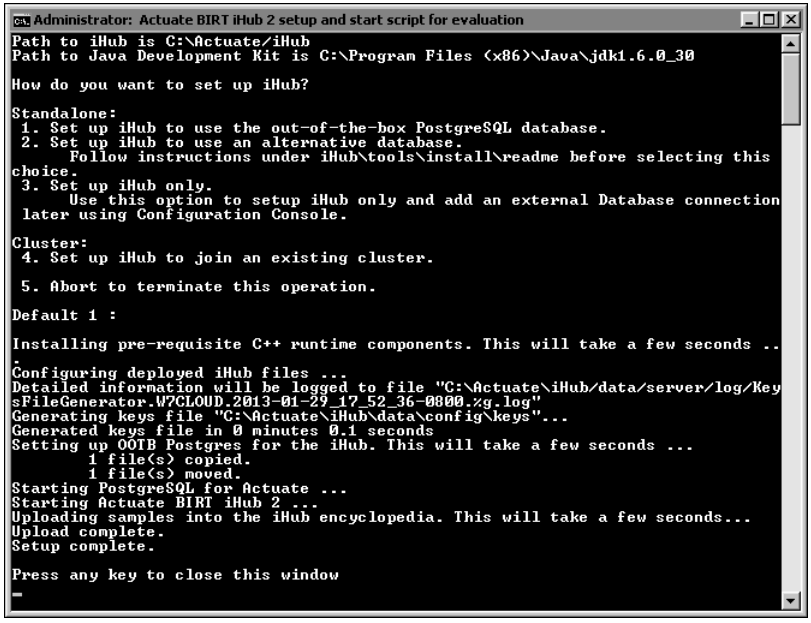


Figure 6-3 Using the setup script to install option 1

When the script starts Actuate BIRT iHub, a command prompt opens to serve as the console from which iHub can be shut down, as shown in Figure 6-4.

Figure 6-4 Viewing Actuate BIRT iHub 2 console

When the script starts PostgreSQL for Actuate BIRT iHub, a command prompt opens to serve as the console from which PostgreSQL can be shut down, as shown in Figure 6-5.

Figure 6-5 Viewing PostgreSQL for Actuate iHub console

For information about starting and stopping Actuate BIRT iHub from a command prompt, see "How to stop and start Actuate BIRT iHub," later in this section. For information about starting and stopping PostgreSQL for Actuate BIRT iHub, see "How to stop and start PostgreSQL for Actuate iHub," later in this section.

If a Windows Security Alert appears indicating that the firewall is blocking access to Actuate BIRT iHub programs, perform the following tasks:

1. In Allow Actuate BIRT iHub to communicate on these networks, for example, select Private networks, such as my home or work network, then choose Allow access, as shown in Figure 6-6.
2. Repeat this step for other Windows Security Alerts, such as Java Platform SE binary.

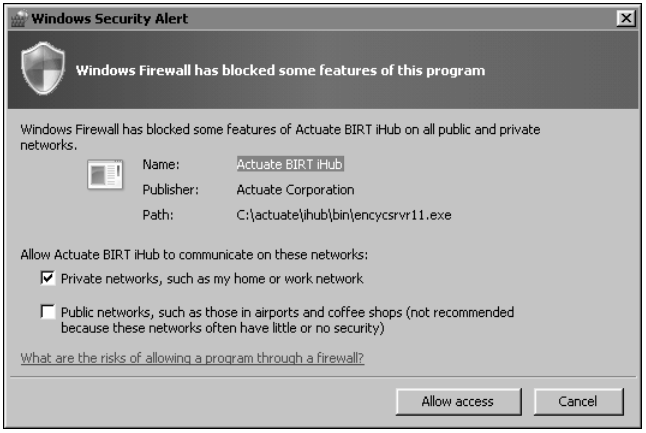


Figure 6-6 Allowing firewall access to Actuate BIRT iHub

When the script finishes running, press any key to close the command prompt running setupiHub.bat script, as shown in Figure 6-3. If the script exit returns to the command line, type exit to close the command prompt. Be careful to not close the command prompt running the Actuate BIRT iHub console or PostgreSQL by mistake.

Accessing Information, Management, and Configuration Consoles

In Windows, the deployment program installs shortcuts to the folder where you extracted the BIRT iHub deployment package. These shortcuts provide access to the following iHub consoles:

- BIRT iHub 2 Information Console
  Launches Information Console to access folders and view designs and documents

- BIRT iHub 2 Management Console

  Launches Management Console to set up user accounts and schedule or run a design

After the script finishes running, a browser opens displaying Welcome to Actuate BIRT iHub, as shown in Figure 6-7.

Log in to Information Console by choosing the shortcut. Alternatively, open a browser manually and enter the following URL, as shown in Figure 6-7:

```
http://localhost:8900/iportal/
```

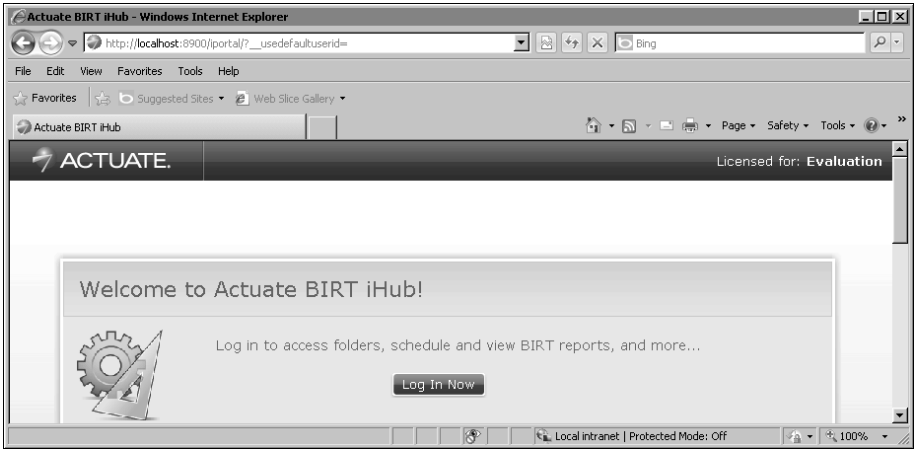


Figure 6-7 Viewing Welcome to Actuate BIRT iHub

Log in to Management Console by choosing the shortcut. Alternatively, open a browser manually and enter the following URL, as shown in Figure 6-8:

```
http://localhost:8900/acadmin/
```

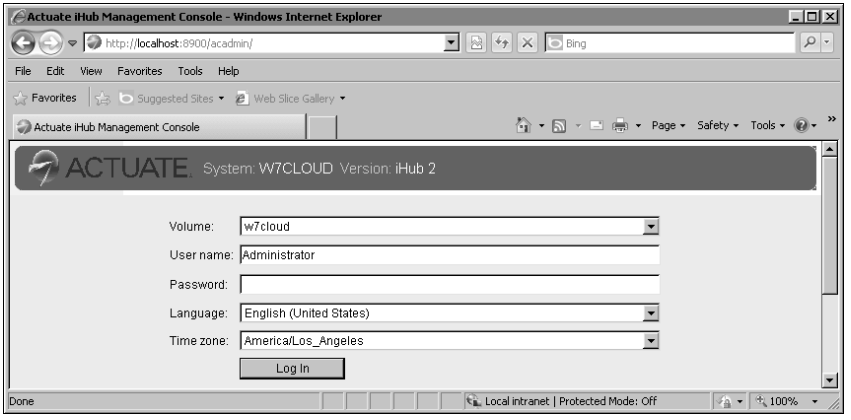


Figure 6-8 Logging in to Management Console

Log in to Configuration Console to perform administrative operations, such as the following tasks:

- Add an Encyclopedia volume.
- Connect to a database.
- Make modifications to iHub parameters and server templates.
- Update the license.

To access Configuration Console for administering iHub, open a browser manually and enter the following URL, as shown in Figure 6-9:

```
http://localhost:8900/acadmin/config/
```

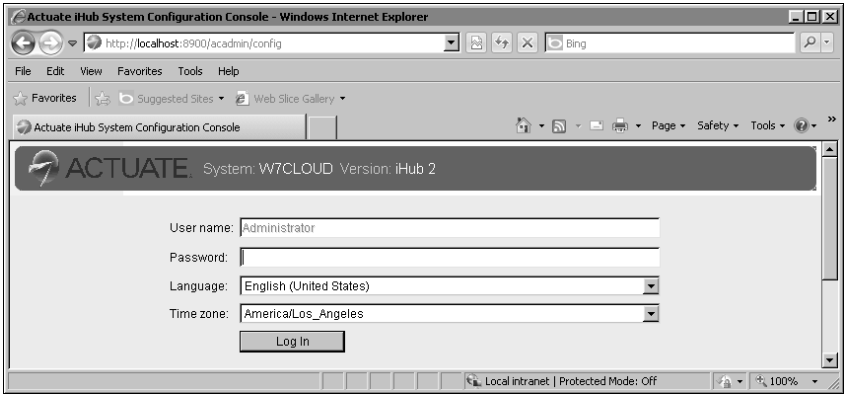


Figure 6-9 Logging in to Configuration Console

When starting PostgreSQL for Actuate iHub and Actuate BIRT iHub, the best practice is to start PostgreSQL then iHub. When stopping these programs, stop iHub then PostgreSQL, if necessary.

It is not necessary to shut down the database when starting and stopping iHub. iHub starts faster with the database already running and available in the background, which is particularly important in cluster and other high-performance operations.

The following sections describe how to perform these operations in the recommended order.

Stopping and starting iHub and PostgreSQL processes

After installing iHub, an administrator can stop and start iHub and PostgreSQL processes from a command prompt using scripts provided in the installation.

How to stop and start Actuate BIRT iHub

To stop Actuate BIRT iHub, perform the following tasks:

1. From the Actuate BIRT iHub console, type s and press Enter.

   The message, Shutting down the server appears, as shown in Figure 6-10.

2. Wait for the shutdown process to complete.

   After shutting down iHub, the command prompt closes automatically.

Figure 6-10 Shutting down Actuate BIRT iHub 2from the console

To restart iHub, perform the following tasks:

1. Open a command prompt and navigate to AC_SERVER_HOME/bin.
2. Type the following command and press Enter, as shown in Figure 6-11:

   ```
   startsrvr
   ```

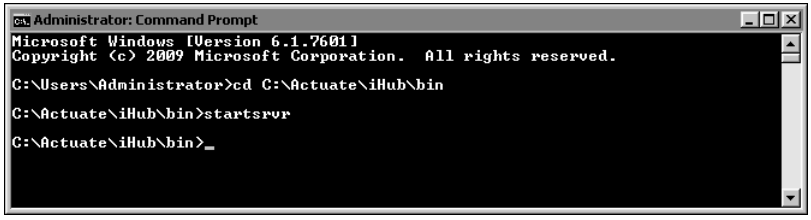


Figure 6-11 Starting Actuate BIRT iHub

A new Actuate BIRT iHub console opens, as shown earlier in Figure 6-4. This window must remain open, since the iHub service runs from this window.

How to stop and start PostgreSQL for Actuate iHub

To shut down PostgreSQL for Actuate iHub from the console that opens during the installation process, shown in Figure 6-5, type Ctrl-C. Wait for the shutdown process to complete. After shutting down PostgreSQL, the command prompt closes automatically.

To restart PostgreSQL for Actuate iHub, perform the following tasks:

1. Open a command prompt and navigate to AC_SERVER_HOME/bin.

2 Type the following command and press Enter, as shown in Figure 6-12:

```
startpostgresql
```

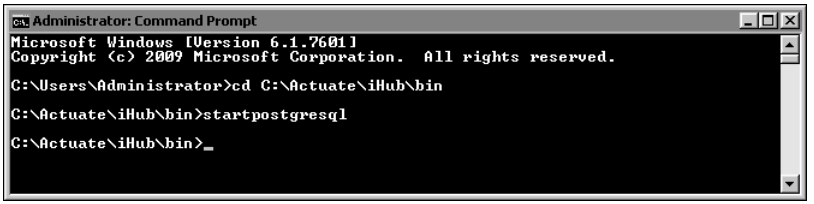


Figure 6-12 Starting PostgreSQL for Actuate iHub

A new command window opens, entitled PostgreSQL for Actuate iHub, as shown in Figure 6-13. This window must remain open, since the iHub service runs from this window.

Figure 6-13 Viewing PostgreSQL for Actuate iHub command window

To stop PostgreSQL, perform the following tasks:

1 Open a command prompt and navigate to AC_SERVER_HOME/bin.

2 Type the following command and press Enter, as shown in Figure 6-14:

```
stoppostgresql
```

The PostgreSQL for Actuate iHub command window closes.

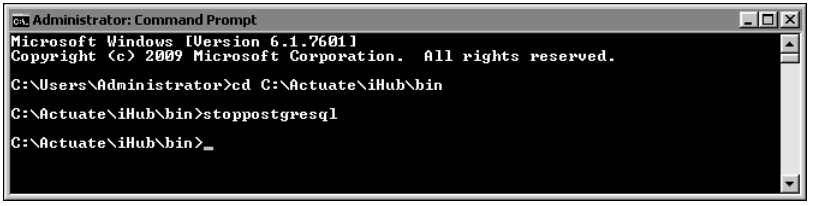


Figure 6-14 Stopping PostgreSQL for Actuate iHub

Setting up iHub to use an alternative database

The following section describes how to install and configure the iHub deployment bundle to use an alternative database.

How to prepare the installation environment

After extracting the contents of ActuateBIRTiHub.zip as described in "How to extract the contents of the iHub distribution package" earlier in this chapter, prepare the installation environment before running the setup script by performing the following tasks:

1. Using Windows Explorer or a command prompt, navigate to AC_SERVER_HOME\tools\install. For example:

   ```
   C:\Actuate\iHub\tools\install
   ```

2. Using a text editor, open the readme.txt file that contains the preliminary setup instructions, as shown in Listing 6-2:

   Listing 6-2 The readme.txt file

   ```
   1. Install database client software onto the iHub node.
   2. Configure the database client to access the database
      instance to which you want to install the encyclopedia.
      In some cases, you may need to set the correct environment
      variables in the command window from which you run the
      script. For example, to run the scripts on Unix with DB2
      as the database, you will need to configure environment
      variables such as DB2INSTANCE, INSTHOME, PATH and
      LD_LIBRARY_PATH to make db2 commands accessible.
   3. Create a "lib" folder under $AC_SERVER_HOME/tools/install.
      Copy JDBC driver jar from database client to this "lib"
      folder.
      - For Oracle database copy ojdbc14.jar
      - For DB2 database, copy db2jcc.jar
      - For SQL Server database, copy sqljdbc4.jar
      - For PostgreSQL database, copy postgresql-8.4-701.jdbc4.jar
   4. Edit the install.properties file to add database connection
      properties and other required properties.
   5. The script is called by other scripts, for example
      startiHub_Non_EmbeddedDB.bat. It can also be executed
      manually as below:
      ant -f install.xml install
   ```

3. Following the instructions in readme.txt, perform the following tasks:

   1. Install an alternative database, such as DB2, Microsoft SQL Server, Oracle, or a pre-existing PostgreSQL server.

   2. Run a SQL script containing the appropriate Data Definition Language (DDL) statements to create the iserver database. For example, when creating the database in a PostgreSQL server, run the following DDL commands:

1. Connect to the PostgreSQL system database as a user with full administrator privileges, typically named postgres, and execute the following SQL commands to create a database named iserver:

```
CREATE DATABASE iserver
   WITH OWNER = "postgres"
   TEMPLATE = template0 ENCODING = 'UTF-8';
REVOKE ALL ON DATABASE iserver FROM PUBLIC;
```

2. In the iserver database, create the plpgsql procedural language by executing the following SQL command:

```
CREATE LANGUAGE plpgsql;
```

   plpgsql is a superset of PostgreSQL that supports advanced programming features, such as variables, conditional expressions, iterative constructs, and events. If the language is already installed, an error message appears. If so, ignore the message.

When the setupiHub script runs, as described in "How to set up iHub to use an alternative database," later in this section, the script creates the following schema owner and application user accounts with appropriate privileges:

- iHub system schema owner
- Encyclopedia volume schema owner
- iserver application user

For more information about creating the iHub system and Encyclopedia volume schemas and iserver user in an alternative database, see Chapter 3, "Installing BIRT iHub using an alternative database."

3. Create a lib folder in AC_SERVER_HOME/tools/install, and copy the JDBC driver JAR file specified for the database to the lib folder.
4. In a text editor, open the install.properties file, and specify all required and any necessary optional settings, as shown in Listing 6-3.

   The install.properties file requires settings for the following properties:

   - AC_SERVER_HOME

     iHub home folder, such as C:/Actuate/iHub. Use forward slashes in the path specification.

   - SYSTEM_NAME

     Set up automatically by the installation script. Do not change this property. Restrict system, schema, and the iHub application user names to alphanumeric and underscore characters with an initial alphabetic character in the pattern [a-z][a-z 0-9]*. Do not use a hyphen.

- DEFAULT_DATABASE _PASSWORD
  Default database server administrator (DBA) or superuser password if the user does not specify a password at the script prompt.
- Database-specific properties:
  - jdbc.dbtype
    Server type, such as DB2, Microsoft SQL Server, Oracle, or PostgreSQL.
  - jdbc.serverName
    Database server name.
  - jdbc.portNumber
    Database server port number.
  - jdbc.databaseName
    Database name, such as iserver.
  - dba.name
    Database administrator (DBA) name.
  - dba.password
    Database administrator (DBA) password. The installation script prompts for the password, so this value can be left at the default setting.

  Listing 6-3 specifies the required property settings for a PostgreSQL database as an example.

Listing 6-3 The install.properties file

```
#Please specify the follow required properties:
   AC_SERVER_HOME, AC_DATA_HOME and AC_CONFIG_HOME
#for the iHub install.  Do not use backslash ("\") in the
   path.  Always use forward slash ("/")
#on both Windows, Linux, or Unix.

#REQUIRED
#AC_SERVER_HOME, for example: D:/ActuateBIRTiHub/iHub
AC_SERVER_HOME=C:/Actuate/iHub

#OPTIONAL
#AC_DATA_HOME, for example: ${AC_SERVER_HOME}/data

#OPTIONAL
#AC_CONFIG_HOME, for example: ${AC_SERVER_HOME}/data/config

#REQUIRED
```

```
#Set up by the script automatically. User should not change
   it.
SYSTEM_NAME=#AC_SYSTEM_NAME#

#REQUIRED
#Database password that Installer asked for, which will be
   applied to all DB related passwords, if user doesn't
   specify.
DEFAULT_DATABASE_PASSWORD=xxxxxx

#OPTIONAL
#APPLICATION_USER_PASSWORD=${DEFAULT_DATABASE_PASSWORD}

#OPTIONAL
#SYSTEM_SCHEMA_PASSWORD=${DEFAULT_DATABASE_PASSWORD}

#OPTIONAL
#VOLUME_SCHEMA_PASSWORD=${DEFAULT_DATABASE_PASSWORD}

#
#Specify database specifc properties. The database types
   supported are Microsoft SQL Server, Oracle, DB2, and
   PostgreSQL.
...
jdbc.dbtype=PostgreSQL
jdbc.serverName=localhost
jdbc.portNumber=8432
jdbc.databaseName=iserver

#define the target database
dba.name=postgres
dba.password=xxxxxx
...
```

How to set up iHub to use an alternative database

After preparing the installation environment, run the setup script as described in "How to run the setup script," earlier in this chapter, and perform the following installation tasks:

1 In the setupiHub menu, choose option 2, Set up iHub to use an alternative database, as shown in Figure 6-15.

The script performs the following tasks, as shown in Figure 6-15:

- Installs the prerequisite C++ run-time components
- Sets up the iHub deployment files, including log and security keys files
- Sets up and starts BIRT iHub

- Creates the Encyclopedia volume
- Uploads the Encyclopedia volume sample content

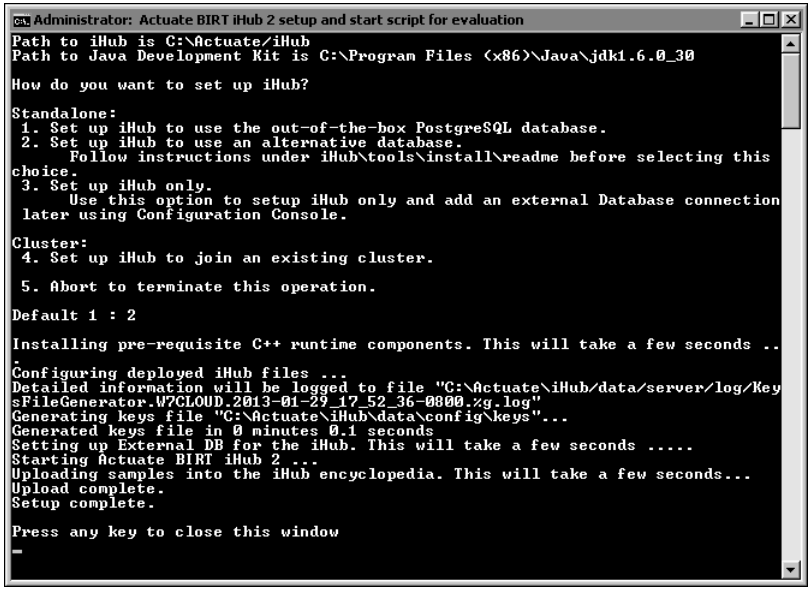


Figure 6-15 Completing the iHub alternative database setup

When the script starts iHub, a new command prompt opens to serve as the Actuate BIRT iHub console from which iHub can be shut down, as shown in Figure 6-16.

When the script finishes running, press any key to close the command prompt running setupiHub.bat script, as shown in Figure 6-15. If the script exit returns to the command line, type exit to close the command prompt. Be careful to not close the command prompt running the Actuate BIRT iHub console by mistake.

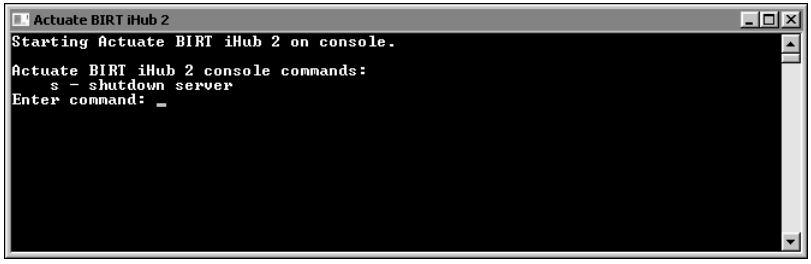


Figure 6-16 Viewing Actuate BIRT iHub 2 console

After the script finishes running, a browser opens displaying Welcome to Actuate BIRT iHub, as shown earlier in Figure 6-7. For more information about accessing iHub consoles, see "Accessing Information, Management, and Configuration Consoles," earlier in this chapter.

Setting up iHub only

The following section describes how to install and configure iHub only and add the external database connections later using Configuration Console. In this option, the administrator performs the following installation tasks:

- Sets up iHub run-time environment only using the cloud deployment package
- Configures an external database to store iHub system and Encyclopedia volume metadata
- Uses Configuration Console to configure a database connection by specifying the properties for the following items:
  - Metadata database
  - System and Encyclopedia volume schemas and iserver application user roles
  - Data partition and Encyclopedia volume

Setting up iHub only using the cloud deployment package

After extracting the contents of ActuateBIRTiHub.zip and running the setup script as described in "How to extract the contents of the iHub distribution package," and "How to run the setup script," earlier in this chapter, perform the following tasks.

How to set up iHub only

1 In the setupiHub menu, choose option 3, Set up iHub only.

The script performs the following tasks, as shown in Figure 6-17:

- Installs the prerequisite C++ run-time components
- Sets up the iHub deployment files, including log and security keys files
- Sets up and starts BIRT iHub 2

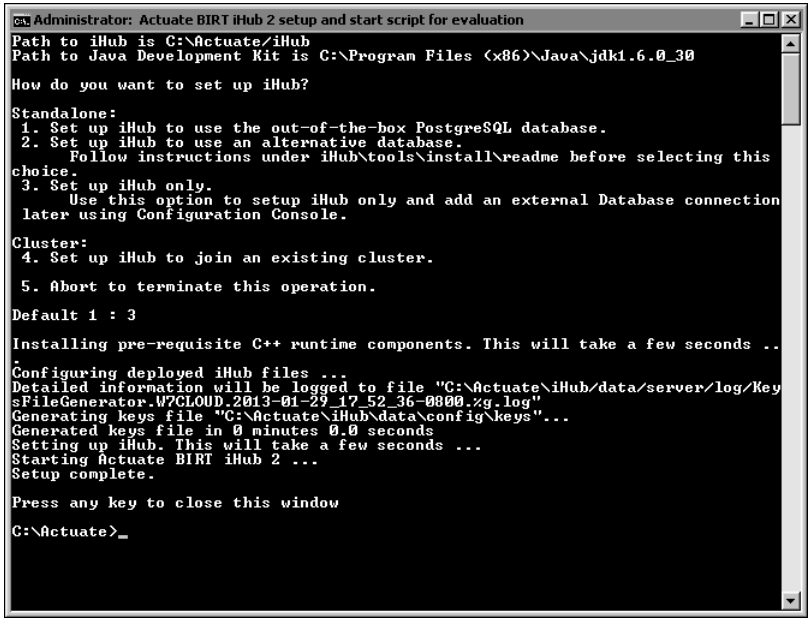


Figure 6-17 Completing the iHub-only setup

When the script starts iHub, a new command prompt opens to function as the Actuate BIRT iHub console from which iHub can be shut down, as shown in Figure 6-18.

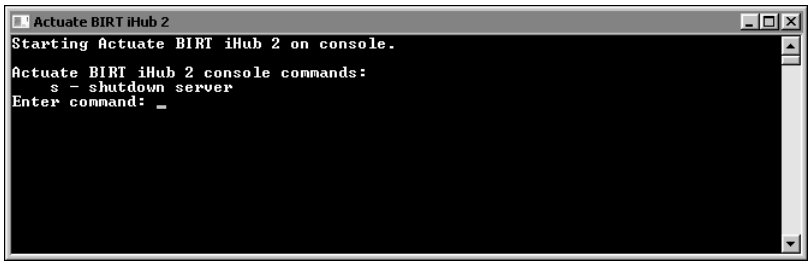


Figure 6-18 Viewing Actuate BIRT iHub 2 console

When the script finishes running, a browser opens displaying Welcome to Actuate Information Console, as shown earlier in Figure 6-7, but no Encyclopedia volume is available. You cannot log in to Information Console until an Encyclopedia volume exists.

2 After the script finishes running, press any key to close the command prompt running setupiHub.bat script, as shown in Figure 6-17. If the script exit returns to the command line, type exit to close the command prompt.

3 After installing option 3, stop and restart iHub by performing the following tasks:

1 In Actuate BIRT iHub console, type s to shut down BIRT iHub.

2 Open a new command prompt and navigate to AC_SERVER_HOME/bin.

3 Type the following command and press Enter, as shown in Figure 6-19:

```
startsrvr
```

Figure 6-19 Starting Actuate BIRT iHub 2

A new Actuate BIRT iHub 2 console opens, as shown earlier in Figure 6-18. This window must remain open, since the iHub service runs from this window. For more information about stopping and starting iHub, see "How to stop and start Actuate BIRT iHub," earlier in this chapter.

Configuring an external database

The database administrator must install an external database then run SQL scripts containing the Data Definition Language (DDL) statements to create the following database objects with appropriate privileges:

- Metadata database
- iHub system schema and user role
- Encyclopedia volume schema and user roles
- iserver application user role

How to add an external database

The following section describes how to add an external database, schemas, and user roles using PostgreSQL server as an example. Listing 6-4 shows an example of a SQL script containing Data Definition Language (DDL) statements that create these objects with appropriate privileges in a PostgreSQL database.

Listing 6-4 SQL Data Definition Language (DDL) script

```
# Run in postgres database

CREATE DATABASE iserver
   WITH OWNER = "postgres"
   TEMPLATE = template0 ENCODING = 'UTF-8';
```

```
REVOKE ALL ON DATABASE iserver FROM PUBLIC;

CREATE ROLE ac_corp_system LOGIN PASSWORD 'password';
GRANT CONNECT ON DATABASE iserver TO ac_corp_system;

CREATE ROLE ac_corp LOGIN PASSWORD 'password';
GRANT CONNECT ON DATABASE iserver TO ac_corp;

CREATE ROLE iserver LOGIN PASSWORD 'password';
GRANT CONNECT ON DATABASE iserver TO iserver;

# Run in iserver database

CREATE LANGUAGE plpgsql;

CREATE SCHEMA ac_corp_system AUTHORIZATION ac_corp_system;
GRANT USAGE ON SCHEMA ac_corp_system TO iserver;

CREATE SCHEMA ac_corp AUTHORIZATION ac_corp;
GRANT USAGE ON SCHEMA ac_corp TO iserver;
```

For more information about creating a metadata database, iHub system and Encyclopedia volume schemas, and iserver user or role in a supported alternative database, such as DB2, Microsoft SQL Server, Oracle, or a pre-existing PostgreSQL database, see Chapter 3, "Installing BIRT iHub using an alternative database."

Using Configuration Console to configure the database connection

After configuring the database and completing the iHub only installation, the iHub administrator must use Configuration Console to complete the iHub system setup by specifying the database connection properties for the following items:

- Metadata database
- iHub System schema
- Encyclopedia volume schema

To create these items, perform the following tasks:

1. Log in to Configuration Console by opening a browser and entering the following URL, as shown in Figure 6-20. The administrator does not have a password yet, so leave Password blank.

   ```
   http://localhost:8900/acadmin/config/
   ```

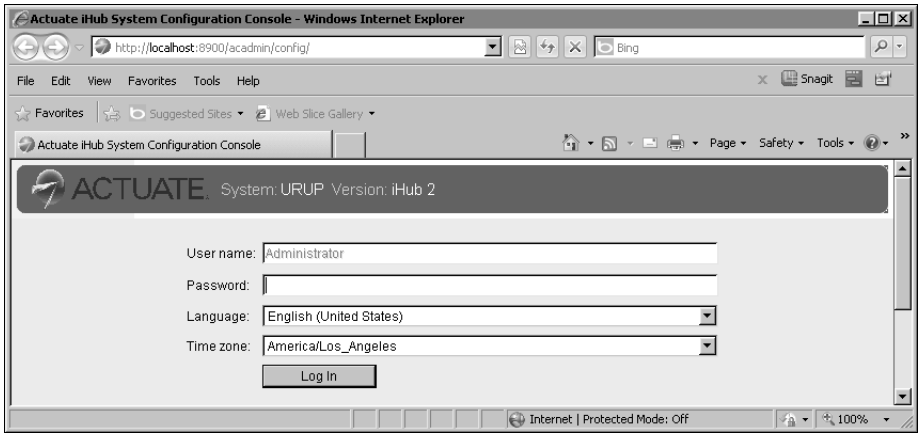


Figure 6-20 Logging in to Configuration Console

2 Choose Advanced view, then choose Volumes.

How to specify a new metadata database

1 To specify a new metadata database, perform the following tasks:

1 Point to the icon next to Metadata Database and choose Add new metadata database, as shown in Figure 6-21.

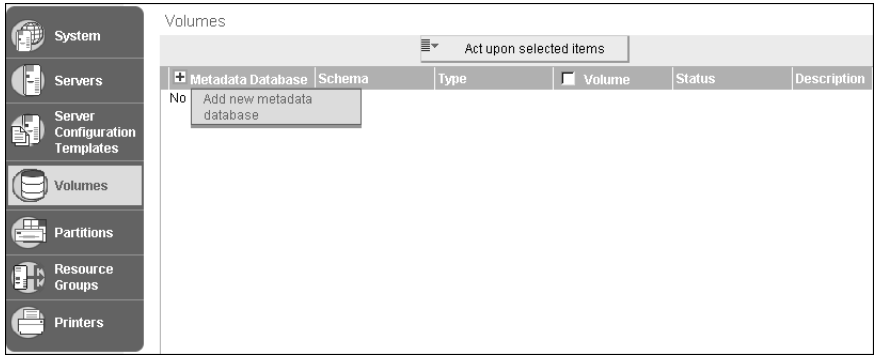


Figure 6-21 Adding a new metadata database

2 On New Metadata Database, perform the following tasks, as shown in Figure 6-22:

1 In Metadata database name, type a name for the metadata database, such as ActuatePostgreSQL_MetadataDatabase.

2 In Database type, select the type of database connection to create, such as PostgreSQL.

3 In Database server, type the host name of the machine containing the database, such as localhost or the actual machine name if the database resides on a remote system.

4 In Database name, type the name for the database, such as iserver.

5 In Connection login, type the database application user name, such as iserver.

6 In Connection password, type the database application user password.

7 In Database port, specify a port number, such as 8432.

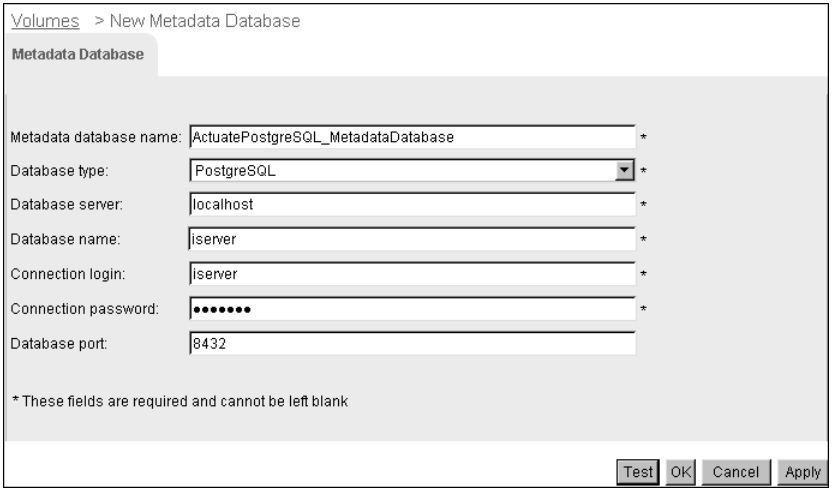


Figure 6-22 Adding new metadata database properties

8 Choose Test.

The message, Connection tested successfully!, appears, if the connection properties are correct and there are no problems communicating with the database system, as shown in Figure 6-23. Choose OK.

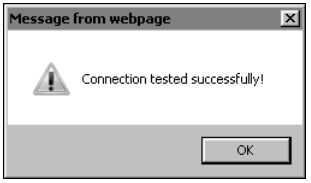


Figure 6-23 Viewing Connection tested successfully! message

On New Metadata Database, choose OK.

9 On Volumes, the ActuatePostgreSQL_MetadataDatabase appears, as shown in Figure 6-24.

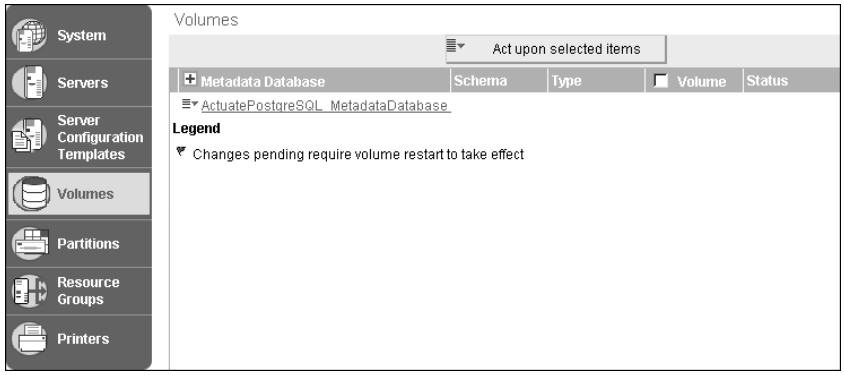


Figure 6-24 Viewing the metadata database

How to specify a new system schema

1 To specify a new system schema, perform the following tasks:

1 On Volumes, point to the icon next to the metadata database and choose Add system schema, as shown in Figure 6-25.

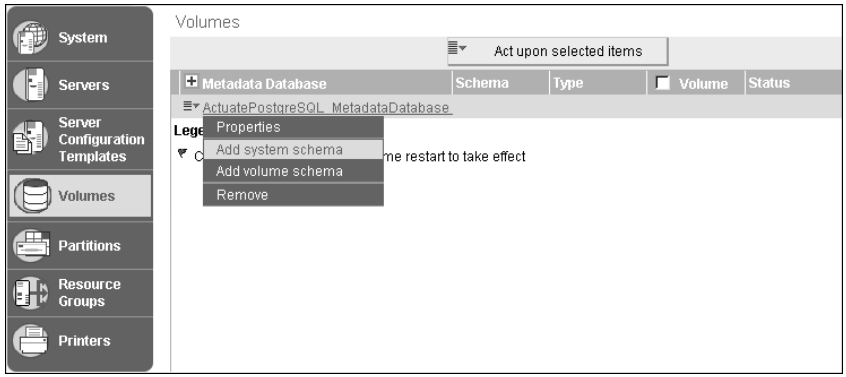


Figure 6-25 Choosing Add system schema

2 On New System Schema, perform the following tasks, as shown in Figure 6-26:

1 In Schema name, type a name for the new schema, such as ac_corp_system. The name must be less than 30 characters.

2 In Schema owner name, type the schema owner name, such as ac_corp_system.

3 Type and confirm a password for the schema owner.

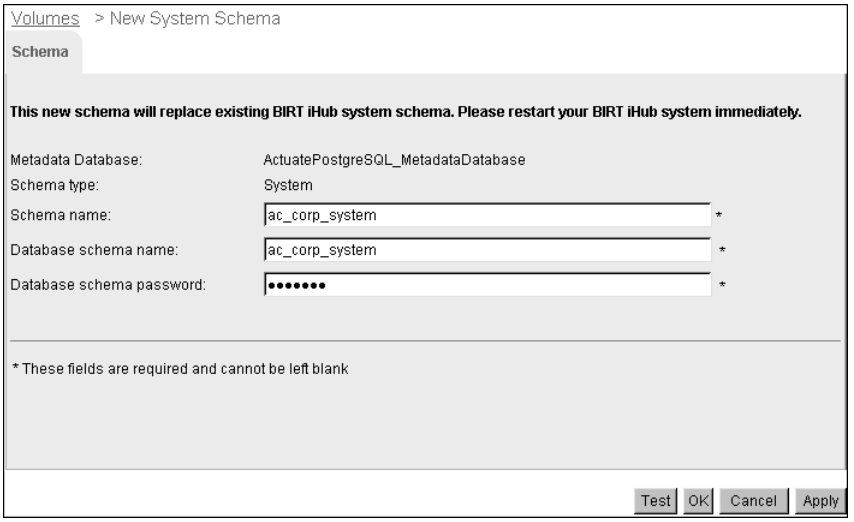


Figure 6-26 Adding a new system schema

4 Choose Test. If successful, choose OK.

5 On New System Schema, choose OK.

A message stating that the new schema will replace the existing BIRT iHub system schema and requesting to restart your system immediately appears, as shown in Figure 6-27.

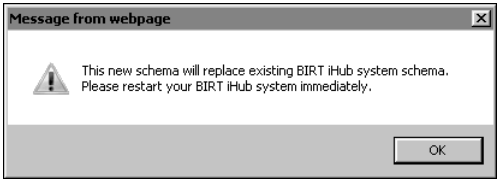


Figure 6-27 Viewing message to restart iHub system

Choose OK.

6 Stop and start iHub as described in "Stopping and starting iHub and PostgreSQL processes," earlier in this chapter.

How to specify a new Encyclopedia volume schema

1 To specify a new Encyclopedia volume schema, perform the following tasks:

1 On Volumes, point to the icon next to the metadata database and choose Add volume schema. The metadata database is ActuatePostgreSQL_MetadataDatabase, as shown in Figure 6-28.

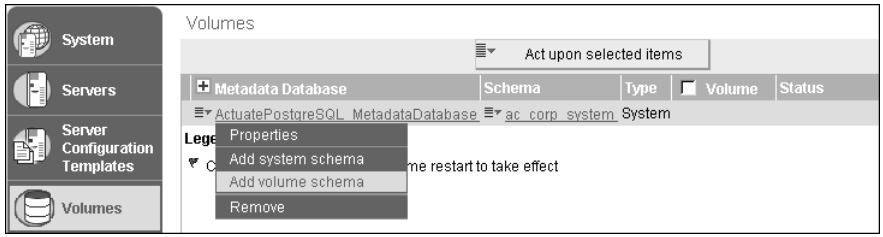


Figure 6-28 Choosing Add volume schema

2 On New Volume Schema, as shown in Figure 6-29, perform the following tasks:

1 In Schema name, type a name for the new schema, such as ac_corp. The name must be less than 30 characters.

2 In Schema owner name, type the schema owner name, such as ac_corp.

3 Type and confirm a password for the schema owner.

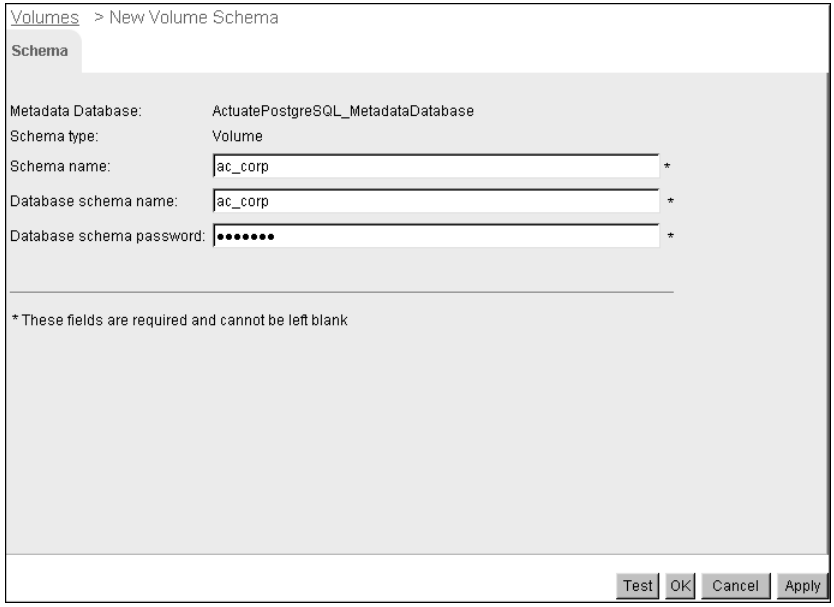


Figure 6-29 Adding a new Encyclopedia volume schema

4 Choose Test.

The message, Connection tested successfully!, appears, if the connection properties are correct and there are no problems communicating with the database system, as shown in Figure 6-30. Choose OK.

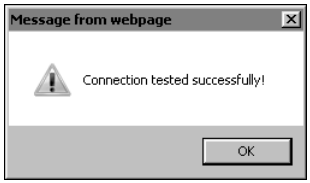


Figure 6-30 Viewing Connection tested successfully! message

On New Volume Schema, choose OK.

Using Configuration Console to configure the data partition and Encyclopedia volume

After configuring the database connection, the administrator must create the following items to bring an Encyclopedia volume online:

- Data partition
- Encyclopedia volume

How to specify a new data partition

In Configuration Console, use the default data partition that installs with iHub or specify a new partition, then create the Encyclopedia volume and bind it to the partition.

1. To delete the default partition and specify a new data partition, perform the following tasks:
   1. Choose Advanced view.
   2. From the side menu, choose Partitions. On Partitions, point to the arrow next to DefaultPartition and choose Delete, as shown in Figure 6-31.

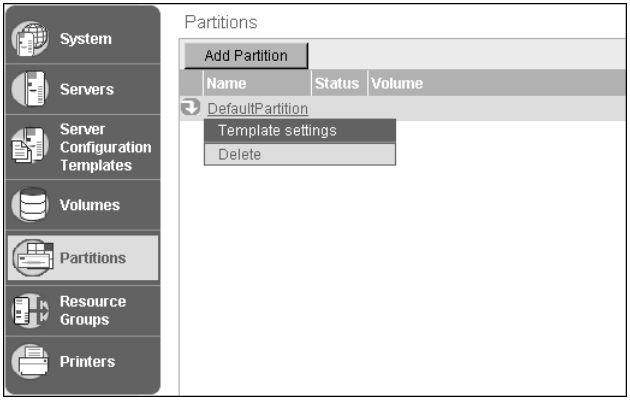


Figure 6-31 Deleting DefaultPartition

Choose OK to confirm the deletion of DefaultPartition, as shown in Figure 6-32.

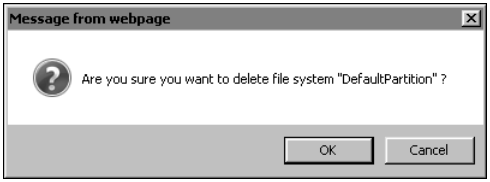


Figure 6-32 Confirming to delete DefaultPartition

2 To specify a new data partition, perform the following tasks:

1 Create a directory for the partition on the physical drive of the machine or storage device that iHub can access. For example, create a directory in the following path:

```
C:\Actuate\iHub\data\ac_corp_partition
```

2 From the advanced view of Configuration Console, choose Partitions.

3 In Partitions, choose Add partition.

4 In Partition name, specify a name. For example, name the partition ac_corp_partition.

5 In Partition Path, specify the fully qualified path to the partition directory, as shown in Figure 6-33.

Choose OK.

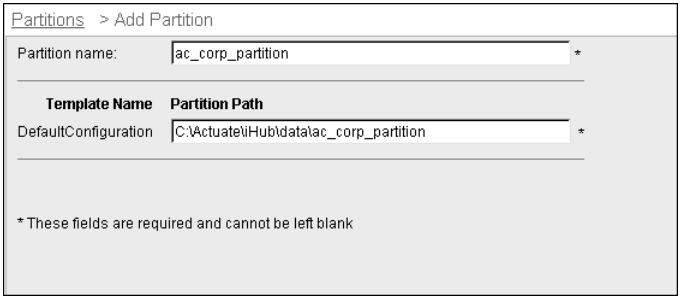


Figure 6-33 Adding a new data partition

6 In Partitions, choose the new partition, ac_corp_partition, from the list of partitions.

7 In Server Settings, choose Test, as shown in Figure 6-34.

Choose OK.

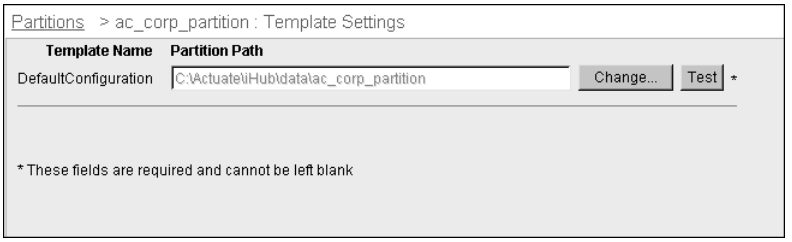


Figure 6-34 Testing a new data partition

If the test succeeds, a success message appears, as shown in message in Figure 6-35. Choose OK.

Figure 6-35 Viewing test successful message

If the test fails, check that the directory named in the partition path exists.

How to specify a new Encyclopedia volume

1 To specify a new Encyclopedia volume, perform the following tasks:

   1 On Volumes, point to the icon next to an Encyclopedia volume schema and choose Add Volume, as shown in Figure 6-36.

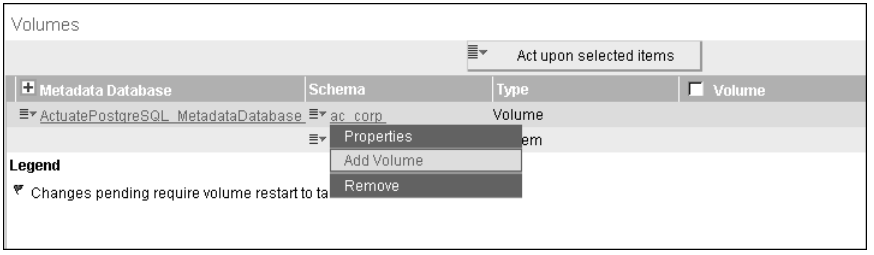


Figure 6-36 Adding a new volume

   2 On New Volume—General, perform the following steps:

      1 Type a name for the new volume. For example, type corp.

      2 In Primary partition, select an unassigned partition. For example, accept ac_corp_partition, as shown in Figure 6-37.

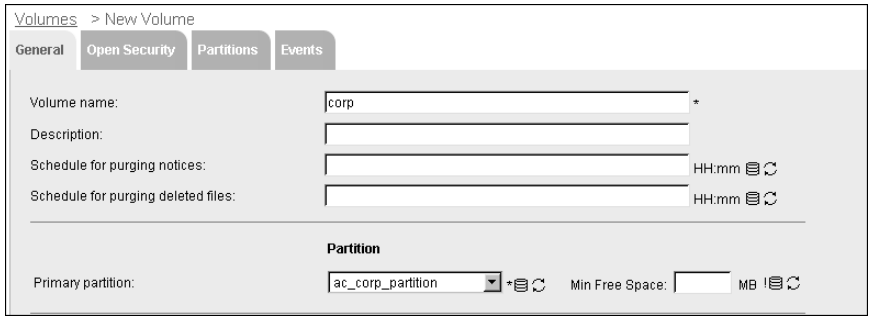


Figure 6-37 Specifying general volume properties

3 On New Volume, choose Partitions, and start the partition for the new Encyclopedia volume by performing the following steps:

1 In Available partitions, select a partition, then move it to Selected by choosing the right arrow.

2 In Selected partitions, select the partition. Choose Start, as shown in Figure 6-38.

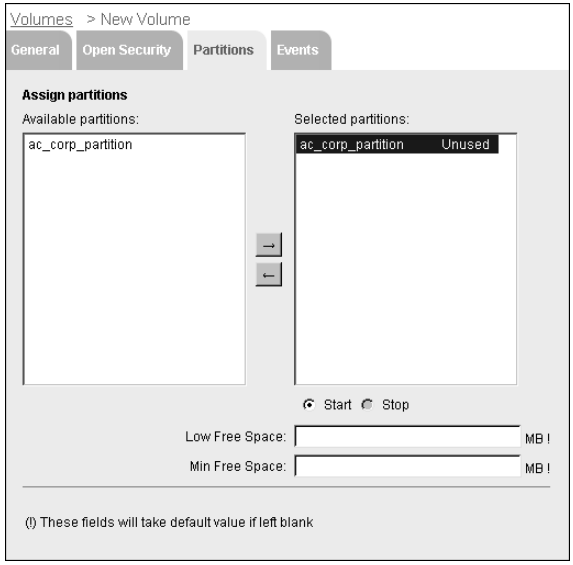


Figure 6-38 Starting the partition

Choose OK.

4 In Volumes, point to the arrow next to the new volume name, and choose Take online, as shown in Figure 6-39.

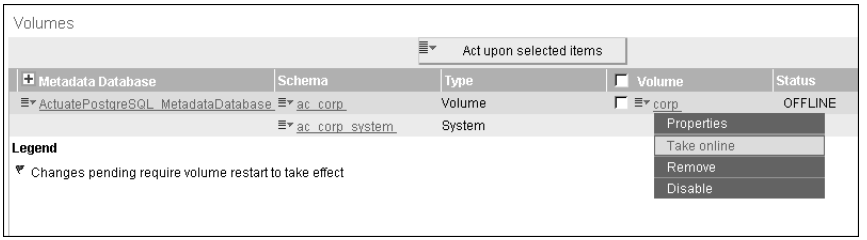


Figure 6-39 Taking a volume online

5 In Volumes, check that the status of the new volume changes to ONLINE, as shown in Figure 6-40.

If the volume does not go online, check for insufficient free disk space for the partition and consider configuring the free space threshold.

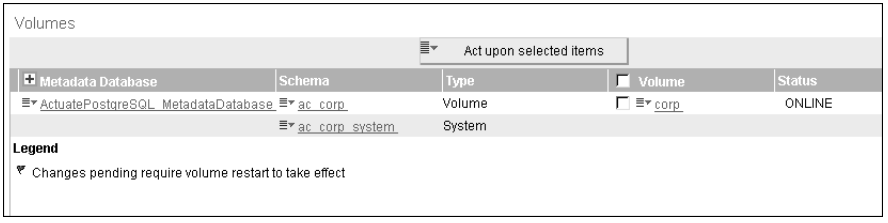


Figure 6-40 Viewing the online volume

6 Log in to Management Console to inspect the new volume by choosing the shortcut. Alternatively, open a browser manually and enter the following URL, as shown in Figure 6-41:

```
http://localhost:8900/acadmin/
```

Figure 6-41 Logging in to Management Console

7 In Files and Folders, the default Encyclopedia volume appears with an empty Resources folder, as shown in Figure 6-42.

Figure 6-42 Viewing new volume contents

Setting up iHub to join an existing cluster

This section assumes the administrator has already created a stand-alone iHub instance on one computer, as described in "How to set up iHub to use the out-of-the-box (OOTB) PostgreSQL database," earlier in this chapter. The stand-alone iHub instance is the machine that contains the shared configuration home directory, which all nodes in the cluster access. This section refers to the machine containing the shared configuration home directory as node1. In "How to set up iHub to join an existing cluster," later in this chapter, the Administrator installs a cluster node. The cluster node accesses the shared resources on node1. This section refers to the machine on which the administrator installs a cluster node as node2.

Before performing a cluster node installation, the administrator performs the following tasks:

- On node1:
  - Turns off the private-network firewall
  - Obtains the machine host name and IP address
  - Tests the network accessibility of the machine
  - Sets the partition path for DefaultPartition in Configuration Console
  - Configures folder sharing and firewall settings in the network environment
- On node2:
  - Turns off the private-network firewall
  - Obtains the machine host name and IP address
  - Tests the network accessibility of the machine

The following instructions provide a basic example of the operations required to configure network sharing and firewall settings in the Windows environment. It is the responsibility of the administrator performing the installation to make sure that all settings conform to the security policies in force for the environment.

How to share the encyc and config folders

Cluster nodes must have read-write access to AC_DATA_HOME\config and AC_DATA_HOME\encyc on node1. To give a cluster node read-write access to the folders and files in the \encyc folder and to the files in the \config folder, perform the following tasks on node1:

1. Using Windows Explorer on node1, right-click the folder, AC_DATA_HOME \encyc. Choose Properties. On encyc Properties, choose Sharing. Sharing appears as shown in Figure 6-43.

Figure 6-43 Sharing the Encyclopedia volume

On encyc Properties, choose Advanced Sharing.

2. On Advanced Sharing, select Share this folder, as shown in Figure 6-44.

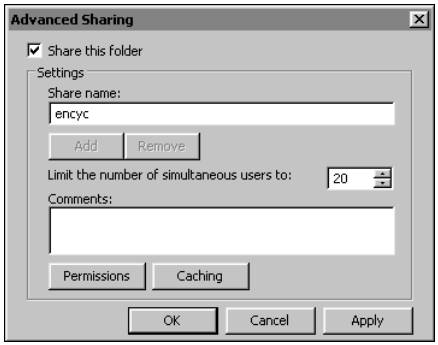


Figure 6-44 Selecting Share this folder

On Advanced Sharing, choose Permissions.

3 On Permissions for encyc—Share Permissions, select the user who installed BIRT iHub. Select Change and Read to allow this user these permissions, as shown in Figure 6-45. Make sure that all settings conform to the security policies in force for the environment.

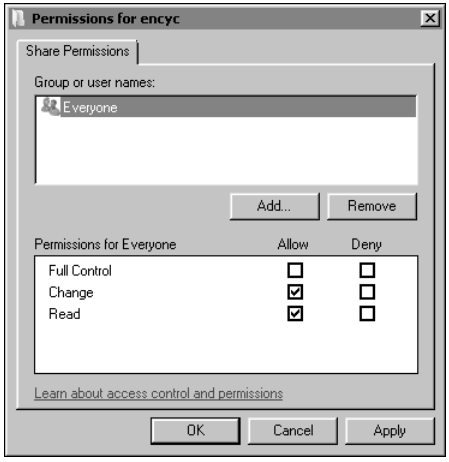


Figure 6-45 Selecting Change and Read permission

Choose OK.

On Advanced Sharing, choose OK.

On encyc Properties, choose Close.

4 Repeat steps 1 through 3 for the \config folder. Make sure that all settings conform to the security policies in force for the environment.

Close Windows Explorer.

How to turn off the Windows firewall

Perform the following steps on node1 and node2:

1. Choose Start➤Control Panel➤System and Security➤Windows Firewall.
2. On Windows Firewall, choose Turn Windows Firewall on or off. Make sure that all settings conform to the security policies in force for the environment.
3. For example, on Customize Settings, in Home or work (private) network location settings, choose Turn off Windows Firewall, as shown in Figure 6-46.

   Choose OK.

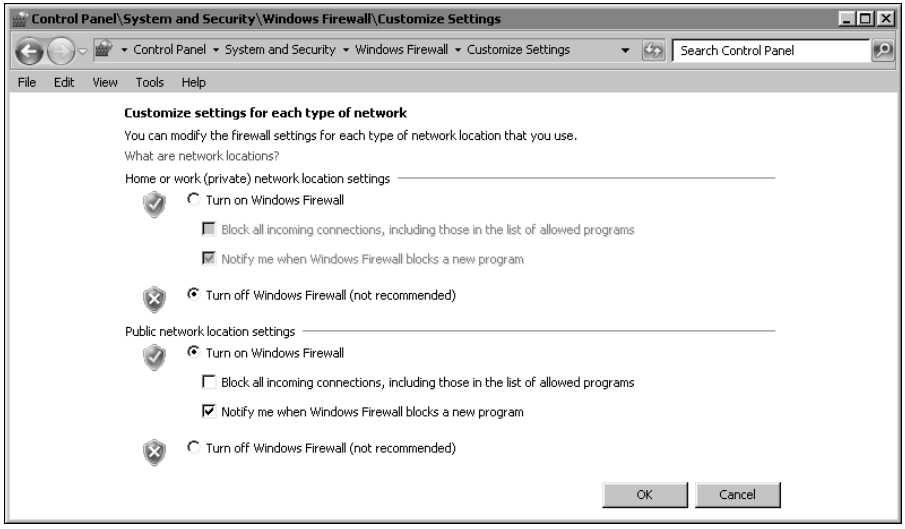


Figure 6-46 Turning off the home or work network location firewall

Close Windows Firewall.

How to display a computer's IP address

To obtain the host names of node1 and the computer on which you will install the cluster node, perform the following tasks on node1 and node2:

1. Choose Start➤Programs➤Accessories➤Command Prompt.
2. In Command Prompt, type the following command:

   ```
   ipconfig /all
   ```

   Press Enter. The host name appears, as shown in Figure 6-47. In this example, the host name for node1 is urup.

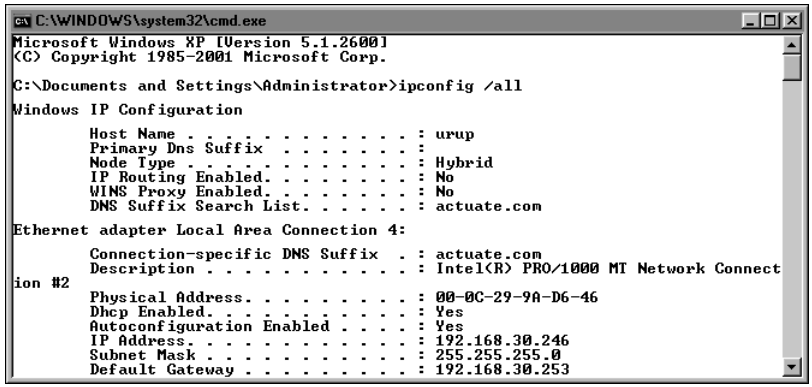


Figure 6-47 Displaying the host name

3 Write down the host names and IP addresses of the computers to be clustered, as shown in Table 6-1.

Table 6-1 Host names and IP addresses of computers to be clustered

iHub	Host name	IP address
Node1	urup	192.168.30.246
Node2	kozu	192.168.30.233

How to test the connection between computers

Perform the following steps on both computers:

1 In Command Prompt, type the ping command followed by the IP address or host name of the other computer. For example, type the following command to ping a computer named kozu:

```
ping kozu
```

Press Enter. If your computer reaches the other computer, Command Prompt displays a series of replies, as shown in Figure 6-48.

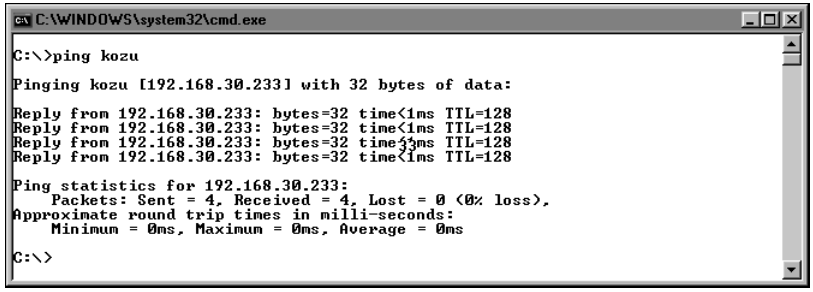


Figure 6-48 Receiving a reply to a ping command

2 Close Command Prompt.

How to prepare node1 for clustering

On node1, or urup in this example, perform the following tasks:

1 Log in to Configuration Console by opening a browser and entering the following URL, as shown in Figure 6-49. The administrator does not have a password yet, so leave Password blank.

```
http://localhost:8900/acadmin/config/
```

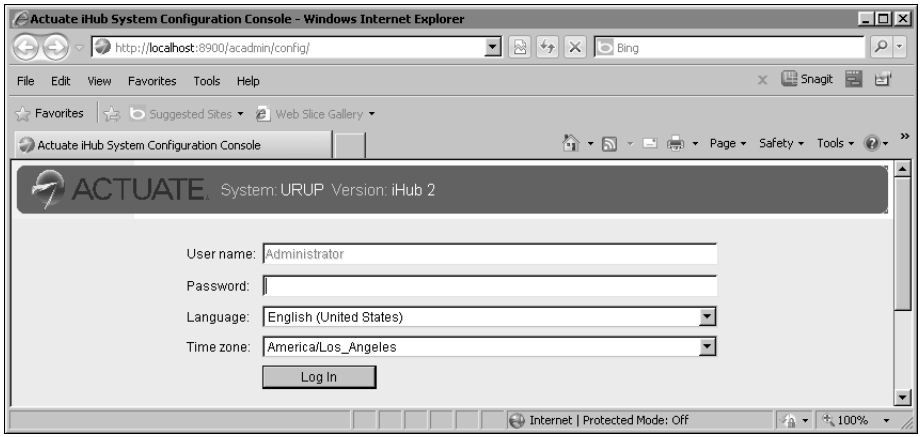


Figure 6-49 Logging in to Configuration Console

2 On the simple view, choose Advanced view. From the side menu, choose Volumes. On Volumes, point to the icon next to the default volume, urup in this example, and choose Take offline, as shown in Figure 6-50.

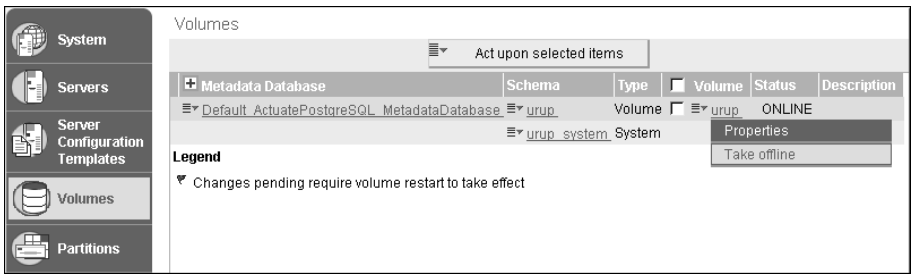


Figure 6-50 Taking a volume offline

In Volume Offline Grace Period, choose OK.

From the side menu, choose Partitions.

3 On Partitions, point to the icon next to DefaultPartition and choose Template settings, as shown in Figure 6-51.

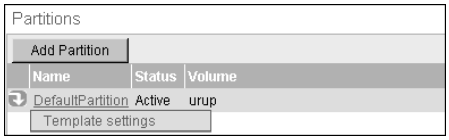


Figure 6-51 Choosing Template settings

On Template Settings, choose Change. In Partition Path, type the path to the encyc folder, using UNC format. For example, if the hostname of node1 is urup, type:

```
\\urup\encyc
```

Template Settings appears as shown in Figure 6-52.

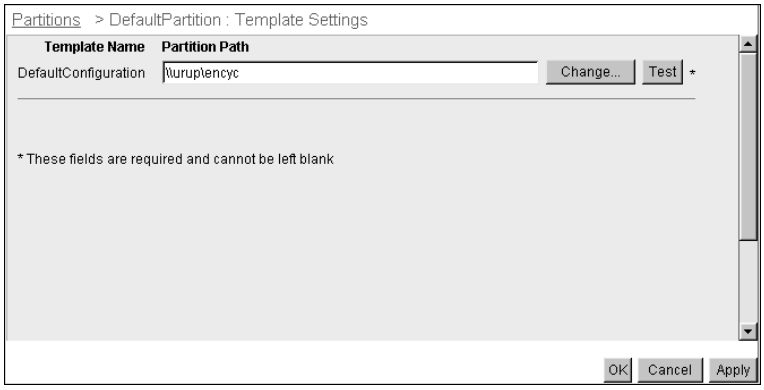


Figure 6-52 Typing the partition path

To verify that iHub can access the encyc folder, choose Test.

A message appears, stating that the test was successful, as shown in Figure 6-53. Choose OK.

Figure 6-53 Verifying a successful partition path test

On Template Settings, choose OK.

4 Log out of Configuration Console.

The following section describes how to install and configure the iHub deployment bundle to join an existing cluster.

How to set up iHub to join an existing cluster

After extracting the contents of ActuateBIRTiHub.zip and running the setup script as described in "How to extract the contents of the iHub distribution package" and "How to run the setup script," earlier in this chapter, perform the following tasks:

1. In the setupiHub menu, choose option 4, Set up iHub to join an existing cluster.
2. At the Cluster config location prompt, specify the configuration home location using Universal Naming Convention (UNC) Format, as shown in Figure 6-54. For example, type:

   ```
   \\urup\config
   ```

   Press Enter.

   In an iHub cluster, the configuration home location, AC_CONFIG_HOME, is the shared directory that contains the acserverconfig.xml, acserverlicense.xml, and other related files. In an iHub installation, by default, these files are in AC_DATA_HOME\config. If the shared folder is in this location on a server named urup, then the administrator specifies \\urup\config as the path.

   For Cluster Template Name, specify a server template name from the available server templates listed in the shared acserverconfig.xml file, as shown in Figure 6-54. For example, type:

   ```
   DefaultConfiguration
   ```

   The default template name in the shared acserverconfig.xml in a cloud-based stand-alone iHub install is DefaultConfiguration.

   Press Enter.

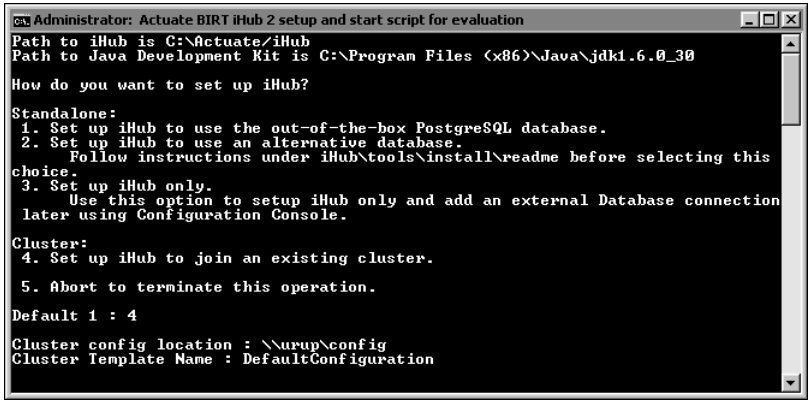


Figure 6-54 Specifying the cluster option, configuration home location, and server template name

The script performs the following tasks, as shown in Figure 6-55:

- Installs the prerequisite C++ run-time components
- Sets up the iHub deployment files, including log and security keys files
- Installs and starts the OOTB PostgreSQL database system used to store Encyclopedia volume metadata
- Sets up and starts BIRT iHub 2

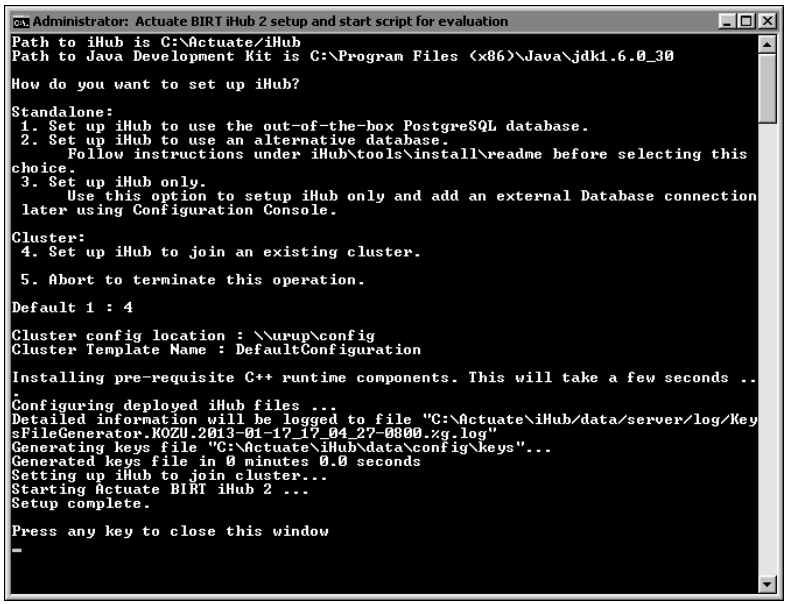


Figure 6-55 Completing the iHub cluster node setup

When the script starts iHub, a new command prompt opens to serve as the Actuate BIRT iHub console from which iHub can be shut down, as shown in Figure 6-56.

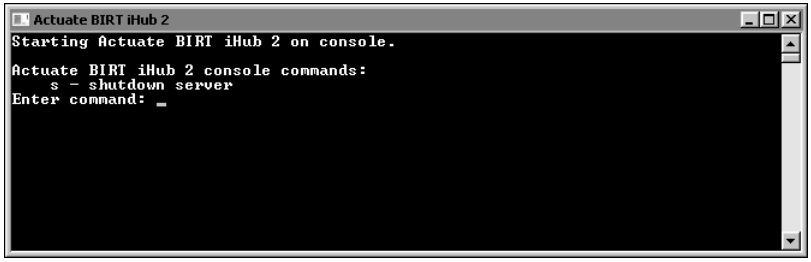


Figure 6-56 Viewing Actuate BIRT iHub 2 console

When the script finishes running, press any key to close the command prompt running the setupiHub.bat script, as shown in Figure 6-55. If the script exit returns to the command line, type exit to close the command prompt. Be careful to not close the command prompt running the Actuate BIRT iHub console by mistake.

In Windows, the deployment program installs shortcuts to the folder where you extracted the BIRT iHub deployment package. These shortcuts provide access to the following iHub consoles:

- BIRT iHub 2 Information Console
  Launches Information Console for viewing report documents
- BIRT iHub 2 Management Console
  Launches Management Console for setting up user accounts and scheduling or running a design

After the script finishes running, a browser opens displaying Welcome to Actuate BIRT iHub, as shown earlier in Figure 6-7. Log in to Information Console to perform tasks such as accessing folders and viewing designs and documents.

3 Log in to Configuration Console. On the simple view, scroll down to Account Settings. In New system password, type a new password. In Confirm system password, type the new password again. Then, choose Change password, as shown in Figure 6-57.

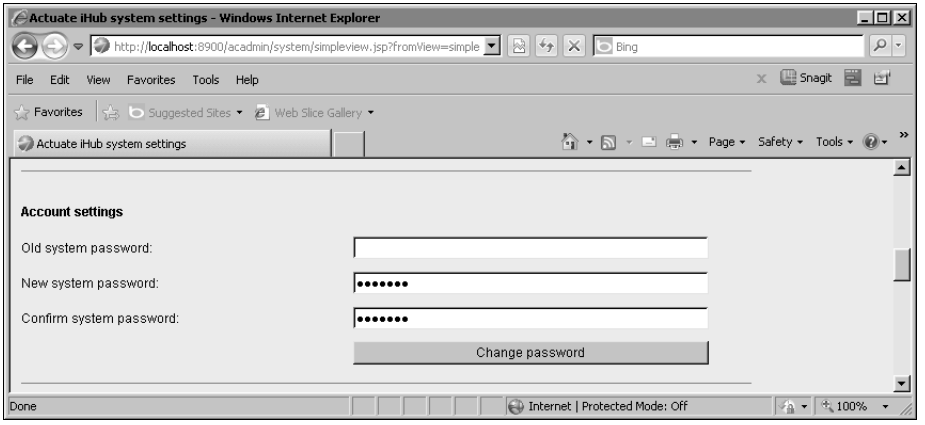


Figure 6-57 Creating a new Configuration Console password

4 Scroll to the top of the simple view and choose Advanced view. In the advanced view, choose Servers. In the example, the node named KOZU has joined the cluster, as shown in Figure 6-58.

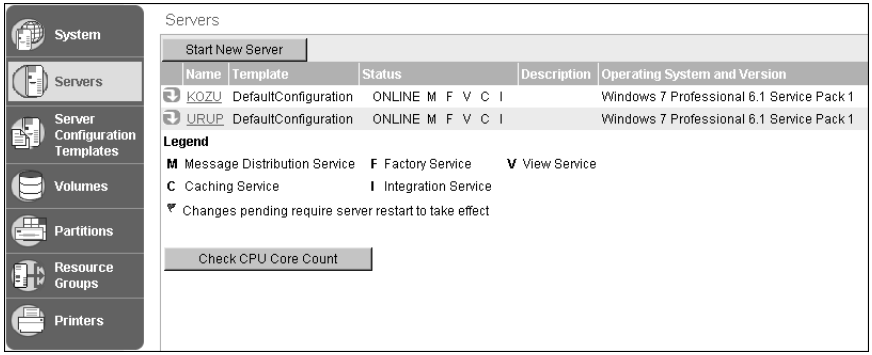


Figure 6-58 Viewing a cluster composed of two nodes

Understanding the cloud deployment environment

In iHub, the location of program files consolidates these resources in AC_SERVER_HOME to facilitate creating an iHub image for deployment in a cloud environment.

iHub uses the environment variable named AC_DATA_HOME for specifying the location of iHub data files. AC_DATA_HOME separates iHub data from the run-time binaries. This change facilitates deployment in a cloud environment.

Specifying AC_SERVER_HOME

The following DLLs, previously installed in WINDOWS\system32 by earlier iServer versions, are now installed in AC_SERVER_HOME\bin directory:

- acicudt18.dll
- acr7790w.dll
- acrs11090.dll
- acxerces-c_1_4_90.dll
- LTDIS10N.dll
- ltfil10N.dll
- ltkrn10N.dll
- mfc90u.dll
- msvcp90.dll
- msvcr71.dll

- msvcr90.dll
- msvcrt.dll

The xmlparse.dll binary, previously installed in AC_SERVER_HOME\operation by earlier iServer versions, is now installed in the AC_SERVER_HOME\bin directory.

The following directories, previously installed under Actuate product home, the parent directory of AC_SERVER_HOME, are now installed under AC_SERVER_HOME directory:

- $AC_PRODUCT_HOME/jar
- $AC_PRODUCT_HOME/MyClasses
- $AC_PRODUCT_HOME/oda

Specifying AC_DATA_HOME

In an iHub product installation, the following data files move to the AC_DATA_HOME directory:

- AC_SERVER_HOME/log
- ACTUATE_HOME/oda/ais/log
- AC_SERVER_HOME/postgresql/log
- AC_SERVER_HOME/server/encyc
- AC_SERVER_HOME/server/encyc/postgresql
- AC_SERVER_HOME/tmp
- AC_SERVER_HOME/etc/acserverconfig.xml and acserverlicense.xml

Other XML configuration files, which are read-only, remain in AC_SERVER_HOME/etc.

Chapter

7

Installing Information Console

This chapter discusses the following topics:

- Preparing to install Information Console
- Installing Information Console on Windows

Preparing to install Information Console

Before installing Information Console, you must prepare the operating system environment to ensure that you have all the necessary software and configuration resources in place. This section describes how to prepare the environment before starting the Information Console installation process.

About installing from an FTP download

If you download an Actuate product from the Actuate FTP site, keep all the files together in their original relative locations when you extract them. The installation program verifies that all necessary files are present before installing Information Console.

If any files are missing, the installation program exits. Files can be missing if you extract the files, move only some of the files to a new location, and attempt to install from that location.

About performing a full installation

In Actuate BIRT iHub System, perform installations of products in the following order:

- Actuate BIRT iHub
- Actuate Information Console
- Actuate BIRT iHub Integration Technology

BIRT iHub installs Configuration, Management, and Information Console automatically on the machine where it resides. Typically, a manual Information Console installation is done to provide network access to iHub from a remote machine.

To access online documentation, such as the online help and PDF files of the product manuals, install the documentation files from the following location:

`http://www.actuate.com`

To install the online help and PDF manuals, from Windows choose Start➤ Actuate➤Update Documentation.

Installing Information Console on Windows

This section describes how to install Actuate BIRT iHub Information Console. Before you begin the installation process, ensure that you have Actuate administrator, system administrator, and web administrator privileges.

Information Console installation requires the following information:

- The application server and port to use. The default settings are the Actuate Apache Tomcat for BIRT iHub Information Console service and port 8700, which the installation program configures. You must configure any other application server. If you are using a firewall, ensure that the firewall allows access to the port number you select.
- The installation wizard installs a Java Development Kit (JDK) distribution and a Java Runtime Environment (JRE) that the Actuate Apache Tomcat for BIRT iHub Information Console service uses.
- The name of the Encyclopedia volume that Information Console accesses. The default is the current machine.

You can install Information Console in two ways:

- Using an installation wizard
  The installation wizard configures Information Console, creates shortcuts, and extracts and installs all necessary files. The installation wizard configures Actuate Apache Tomcat for BIRT iHub Information Console service.
- Deploying a web archive (WAR) file
  Deploying directly requires that you configure for the application server. Use this option if your application server supports configuration of an application from a WAR file.

Using the installation wizard

The following section describes how to install Information Console using the installation wizard.

How to install using the wizard

1. Download the Information Console distribution package from an FTP software distribution site. Extract the files. Run the self-extracting executable file, ActuateInformationConsole.exe. The welcome message appears, as shown in Figure 7-1. Choose Next.

Figure 7-1 Viewing the welcome message

2 Read and accept the license agreement, as shown in Figure 7-2. Choose Next.

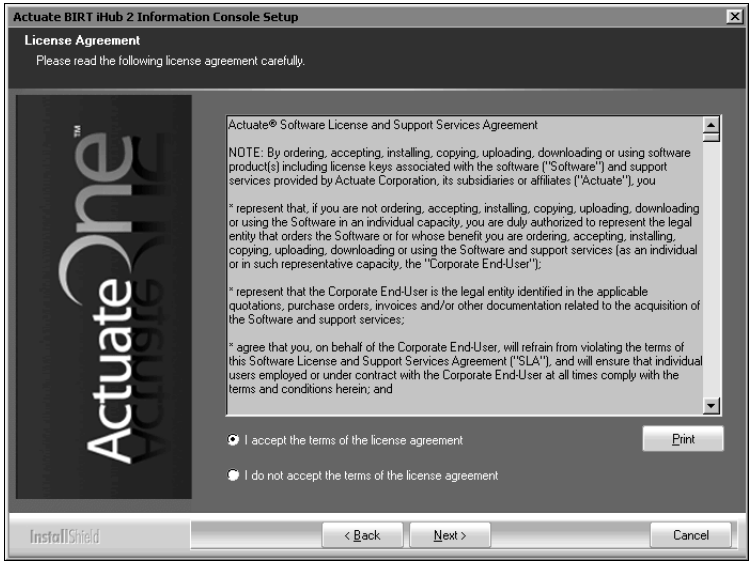


Figure 7-2 Accepting the license agreement

3 In Setup Type, select Typical, as shown in Figure 7-3. Choose Next.

Figure 7-3 Specifying the typical or custom setup type

4 In Locale Information, choose Next to accept the default language and time zone, as shown in Figure 7-4, or specify these settings for your region.

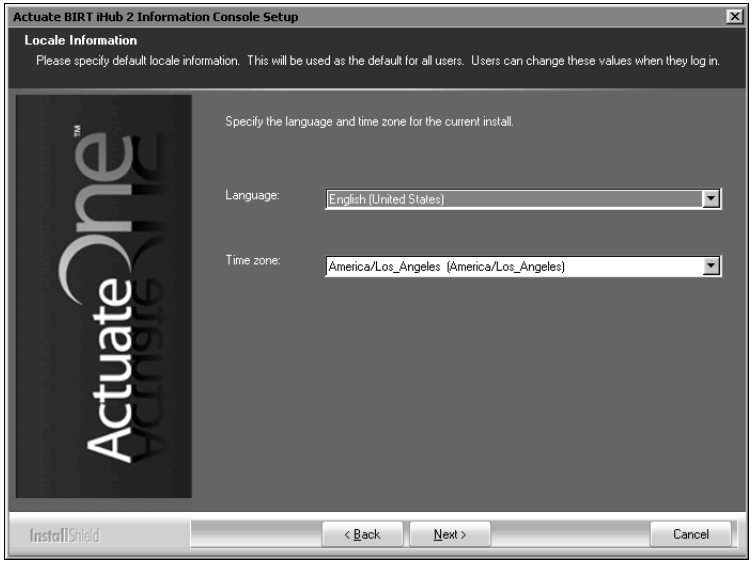


Figure 7-4 Specifying locale information

5 In Apache Tomcat for Actuate Information Console Service, accept the default port, 8700, and context path, /iportal, as shown in Figure 7-5. Choose Next.

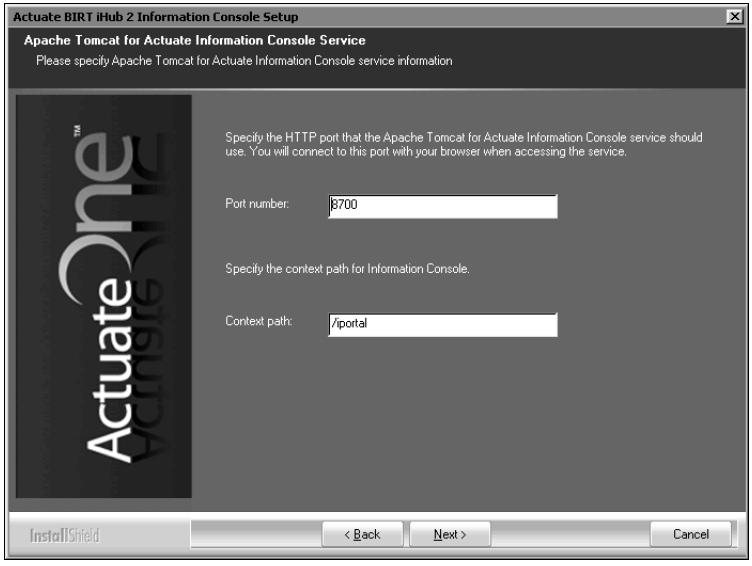


Figure 7-5 Specifying the port number and context path

6 In Specify Windows Account Information, type the user name and password for the account used to run the Actuate Apache Tomcat for BIRT iHub 2 Information Console service, as shown in Figure 7-6. Choose Next.

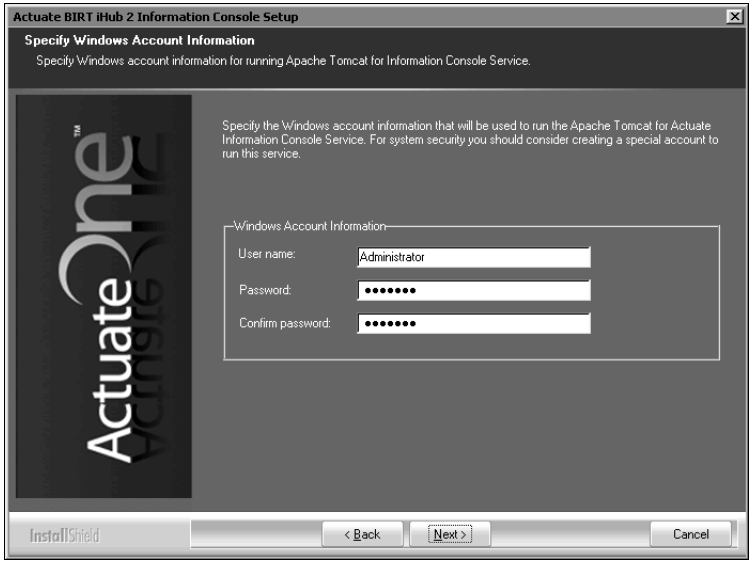


Figure 7-6 Specifying the account running the Information Console service

If prompted to add Log on as a service privilege, choose Yes.

7 In iHub Information, accept the default values for Profile name, Host name, and Port number, as shown in Figure 7-7. Alternatively, type a different value for one or more of these properties. In Volume name, type a name for the default Encyclopedia volume. Choose Next.

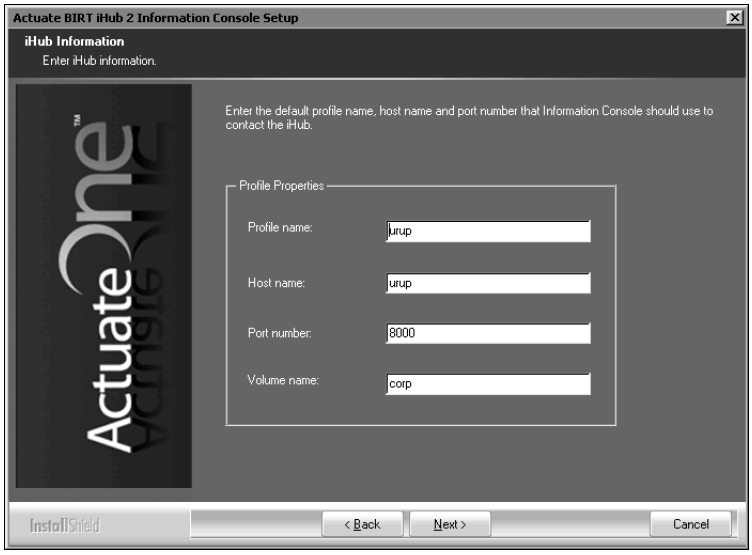


Figure 7-7 Specifying values for iHub profile properties

8 In Start Copying Files, review the settings shown in Figure 7-8. Choose Next.

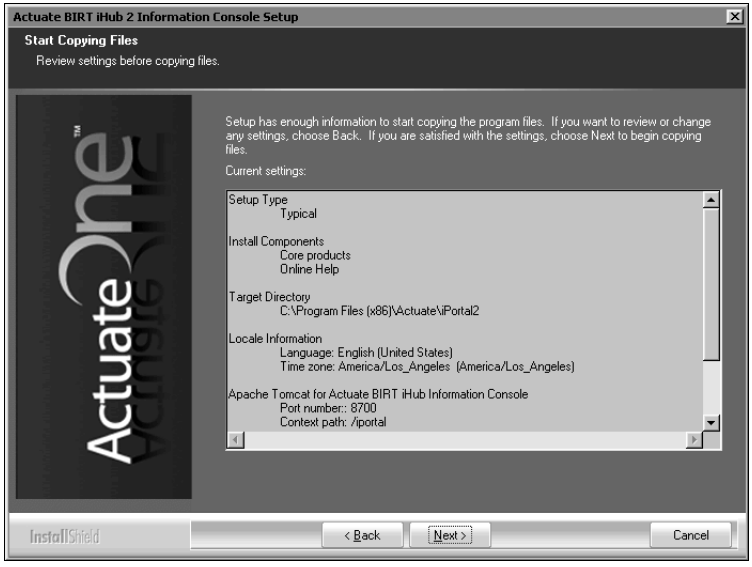


Figure 7-8 Reviewing settings before copying files

Setup Status displays an indicator showing how the installation is progressing, as shown in Figure 7-9.

Figure 7-9 Setup Status

9 Select I would like to view the ReadMe file, as shown in Figure 7-10, if you want to review this documentation. Choose Finish to exit the wizard.

Figure 7-10 Exiting the installation wizard

If you chose to view the ReadMe file, the installation program opens the document, as shown in Figure 7-11.

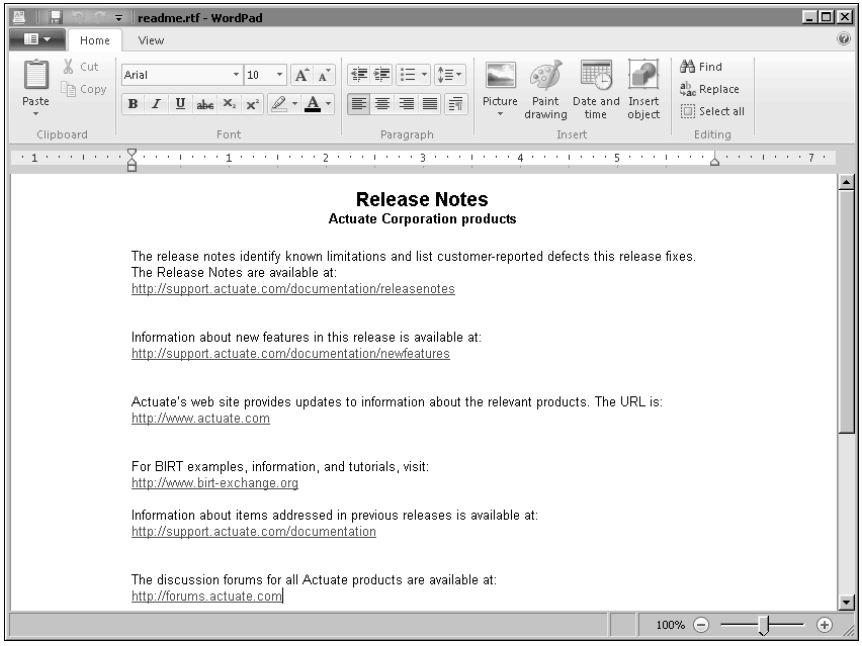


Figure 7-11 The ReadMe file

Using the WAR file to install

If Actuate supports your application server, you can deploy Information Console as a WAR file. See your application server documentation about how to deploy a WAR file. For information about application servers on which Actuate supports deployment of Information Console, see the Supported Products Matrix for this release on the Actuate Support site at the following URL:

`http://support.actuate.com/documentation/spm`

Customize Information Console for your local environment, if necessary, before beginning deployment. To customize the application for your local environment, follow the steps in "Preparing the WAR file," later in this section. Deploy the customized Information Console WAR file to the application server instead of the ActuateInformationConsole.war file that ships with Information Console.

General deployment tasks

You must configure the Information Console WAR and application server for integration. You must also verify that you have a standard J2EE installation.

To integrate your application server with Actuate, complete the following tasks:

- Configure the application server to operate with Actuate products.
- Configure the Information Console WAR to integrate with the application server as described in "Preparing the WAR file," later in this chapter.
- Deploy Information Console to the application server.
  - If the application server has deployment tools, use those tools to integrate the application server with Actuate products. For more information about deployment tools, see your application server's documentation.
  - If your application server does not have deployment tools, add the Actuate context root to your application server, typically by modifying the application server's configuration file.

Preparing the server

Actuate recommends the following configuration for best performance:

- Use at least a two-CPU machine for Information Console.
- If BIRT iHub system uses more than four CPUs, configure one CPU for Information Console for every two CPUs that iHub uses.

Set the following configuration values on the application or web server:

- On a machine with 1 GB of physical memory, set the Java heap size to a value between 256 MB and 512 MB.
- Set the number of threads to a value between 50 and 75.

Preparing the WAR file

You can use a WAR file to deploy Information Console to many supported servers. Table 7-1 describes the Information Console configuration parameters to review and update before deployment.

Table 7-1 Information Console configuration parameters

Parameter name	Description	Action
BIRT _RESOURCE _PATH	The location of the standard templates and properties files that BIRT Studio uses. This location can be in a WAR file or on a disk.	If you specify a location on disk, copy the contents of the Information Console resources folder to this physical location on the file system.
DEFAULT _LOCALE	The default locale is en_US. You can leave this value unchanged. A user can select a locale at login.	If you change the locale, select the new locale from the locales in /WEB-INF/Localemap.xml.

Table 7-1 Information Console configuration parameters

Parameter name	Description	Action
DEFAULT _TIMEZONE	The default time zone is Pacific Standard Time (PST). You can leave this value unchanged. A user can select a time zone at login.	If you change the time zone, select the new time zone from the time zones in the TimeZones.xml file, in the WEB-INF directory.
SERVER _DEFAULT	This value specifies the iHub URL to which the Information Console application connects if you do not specify a server URL. The default value is http://localhost:8000	Update this value to the machine and port of the server. Change localhost to the iHub machine name or IP address. Change 8000 to the iHub port number.
DEFAULT _VOLUME	This value specifies the default Encyclopedia volume for Information Console. If you do not specify a volume in an Information Console URL, the JSP application attempts to log in to this volume. The default value is localhost.	Update this value to the name of an Encyclopedia volume.
BIRT_VIEWER _LOG_DIR	The location of the files that log Actuate BIRT viewer activity.	You must create this physical location on the file system.
LOG_FILE _LOCATION	The location of the files that log Information Console activity.	You must create this physical location on the file system.
TEMP_FOLDER _LOCATION	The location where Information Console creates temporary files.	You must create this physical location on the file system.
TRANSIENT _STORE_PATH	The location where Information Console creates temporary files.	You must create this physical location on the file system.

How to customize the WAR file

The following steps describe the general procedure for customizing an Information Console WAR file:

1. Create a temporary directory, such as C:/Temp/ic.

   If you use an existing directory, ensure that this directory is empty.

2. Open a command window and type the following commands, replacing the X: drive letter with a drive letter appropriate to your system:

   1. Copy the appropriate Information Console WAR file for your environment, such as TOMCAT_ActuateInformationConsole.war, to the temporary directory, as shown in the following example:

      ```
      cd C:/Temp/ic
      copy X:/TOMCAT_ActuateInformationConsole.war .
      ```

2 Decompress the file, as shown in the following example:

```
jar -xf TOMCAT_ActuateInformationConsole.war
```

The Information Console files appear in the temporary directory. Leave the Command window open.

3 Using a text editor that accepts UTF-8 encoding, edit web.xml to configure Information Console for the application server.

If you used the temporary path in step 2, the file location is C:/Temp/ic /WEB-INF/Web.xml. Refer to Table 7-1 for a list of entries to modify in web.xml.

4 Save and close web.xml.

5 Type the following command:

```
jar -cf ../newinformationconsole.war *
```

This command creates newinformationconsole.war in the /temp directory. This new WAR file for Information Console contains the modified configuration values.

Use newinformationconsole.war to deploy Information Console to your application server.

About clusters of servers

If the application server supports clustering, see the application server documentation for more information about setting up clusters and deploying web applications such as Information Console.

Avoiding cache conflicts after installing

Information Console uses Java Server Page (JSP) technology. Application servers and browsers cache pages. A browser can use a cached copy of a page instead of the new page. After you install, using a cached copy of some pages can lead to errors or missing functionality.

To avoid this problem, clear the application server cache after you install Information Console. With some application servers, you must restart the application server. For more information about the necessary steps for clearing the cache, see the application server documentation.

If possible, also clear the browser cache to ensure that the browser does not use an old copy of the page from its cache. Alternatively, you can refresh recently visited pages or clear the browser cache if a problem occurs. For more information about clearing the browser's cache and refreshing a page, see the browser documentation.

Testing the installation

Complete the steps in the following section to test the Information Console installation.

How to test the installation

1. Start the Apache Tomcat for Actuate Information Console service or the application server and web server, if necessary.
2. Start Information Console:
   - If you used the wizard installation, Choose Start➤All Programs➤Actuate➤Information Console.
   - If you used the deployment installation, type a URL in your web browser. For example, type a URL similar to the following example:

     http://Actuate1:8900/iportal/getfolderitems.do?repositoryType=Enterprise&volume=volume1&serverurl=http://iHub1:8000

     where

     - Actuate1:8900 is the name of your computer and the port you use to access Information Console.
     - iportal is the context root for Information Console.
     - ? indicates the beginning of a parameter that indicates where to access Information Console files.
     - getfolderitems.do is the call to the default Information Console home page.
     - repositoryType=Enterprise indicates that this Information Console connects to iHub.
     - &volume=volume1&serverurl=http://iHub1:8000 specifies the Encyclopedia volume and the URL to the BIRT iHub.

   The Information Console login page appears.
3. On the Information Console login page, type a user name and password. Accept the default settings in Encyclopedia Volume, Language, and Time zone.
4. Choose Log In.

Chapter

8

Installing iHub Integration Technology and documentation

This chapter discusses the following topics:

- Installing iHub Integration Technology
- Installing the localization and documentation files
- About accessing online help
- Searching PDF manuals using master-index.pdx

Installing iHub Integration Technology

This section describes how to install iHub Integration Technology for Windows. In a default installation, iHub Integration Technology installs in C:\Program Files (x86)\Actuate\ServerIntTech2.

How to install

To install iHub Integration Technology, perform the following tasks:

1. Download the iHub Integration Technology distribution package from an FTP software distribution site. Extract the files. Run the self-extracting executable file, ActuateiHubIntegrationTechnology.exe. The welcome message appears, as shown in Figure 8-1. Choose Next.

Figure 8-1 Viewing the welcome message

2. Read and accept the license agreement, as shown in Figure 8-2.

   Choose Next.

Figure 8-2 Accepting the license agreement

3 In Setup Type, shown in Figure 8-3, select Typical. Choose Next.

Figure 8-3 Selecting a typical installation

4 In Start Copying Files, review the settings shown in Figure 8-4. Choose Next.

Figure 8-4 Reviewing settings before copying files

Setup Status displays an indicator showing how the installation is progressing, as shown in Figure 8-5.

Figure 8-5 Viewing Setup Status

5 When the setup success message appears, select I would like to view the ReadMe file, as shown in Figure 8-6, if you want to review this documentation. Choose Finish to exit the wizard.

Figure 8-6 Exiting the installation wizard

If you chose to view the ReadMe file, the installation process opens the document, as shown in Figure 8-7.

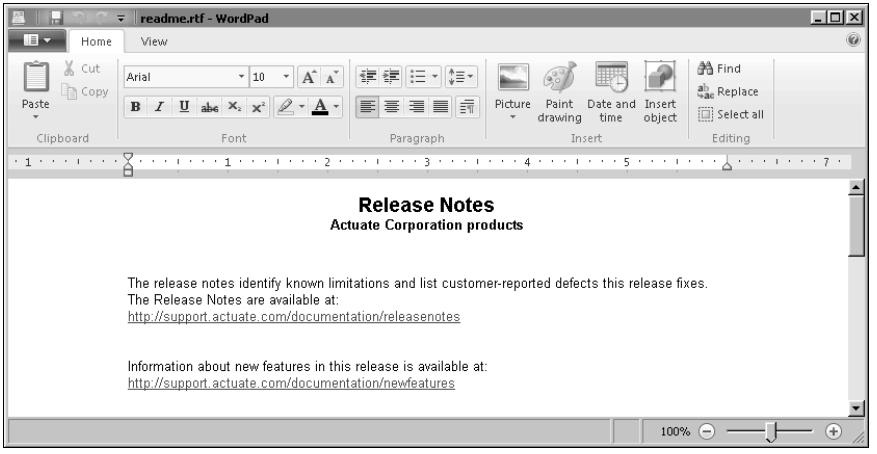


Figure 8-7 Viewing the ReadMe file

6 The installation program prompts you to install the online help from the following location, as shown in Figure 8-8:

```
http://www.actuate.com
```

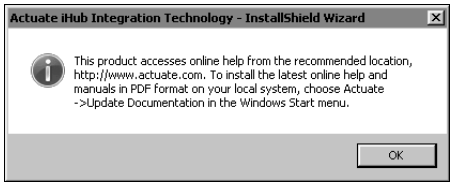


Figure 8-8 Viewing the install online help and manuals prompt

To install the online help and PDF manuals from this location, in Windows choose Start➤ Actuate➤Update Documentation.

Installing the localization and documentation files

The information in the printed manuals is available as Adobe Acrobat PDF files and as a context-sensitive help system for Actuate products. Actuate products access HTML-format documentation from the Actuate web site. If you do not have web access or prefer to use documentation hosted on your local system, install the online documentation files after installing Actuate products.

Both localization and documentation resource file updates can become available between releases. The Actuate Localization and Online Documentation Update tool provides replacement files and additional files for PDF documentation, online help, and localization of installed Actuate products. The tool is available from the Actuate web site, at the following URL:

http://www.actuate.com/docupdateihub2/docupdate.html

Updated documentation in both HTML and PDF form is available at the following Actuate web site location:

http://www.actuate.com/documentation

How to install the localization and documentation files

To install the iHub localization and documentation files, perform the following tasks:

1 Download the Actuate Localization and Documentation distribution package from an FTP software distribution site. Extract the files. Run the self-extracting executable file, ActuateLocalizationandOnlineDocumentation.exe. The welcome message appears, as shown in Figure 8-9. Choose Next.

Figure 8-9 Viewing the welcome message

2 Read and accept the license agreement, as shown in Figure 8-10. Choose Next.

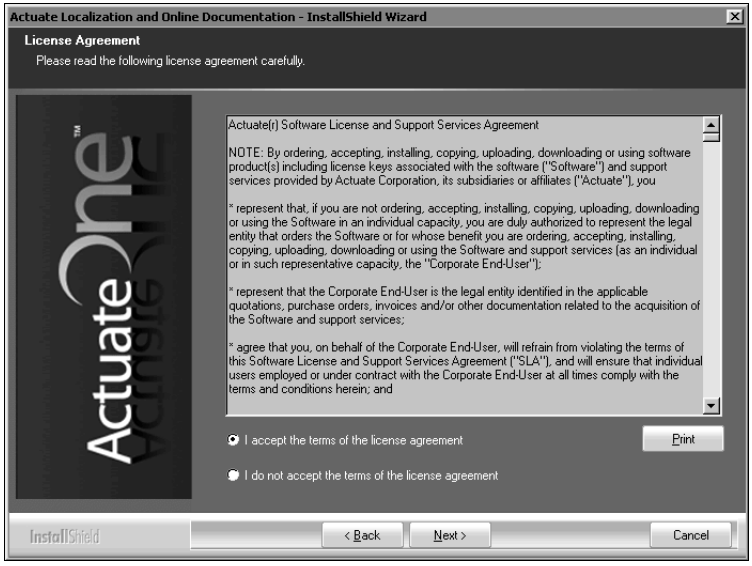


Figure 8-10 Accepting the license agreement

3 In Setup Type, select Typical, as shown in Figure 8-11, or select Custom to specify a limited set of localization and documentation files. Choose Next.

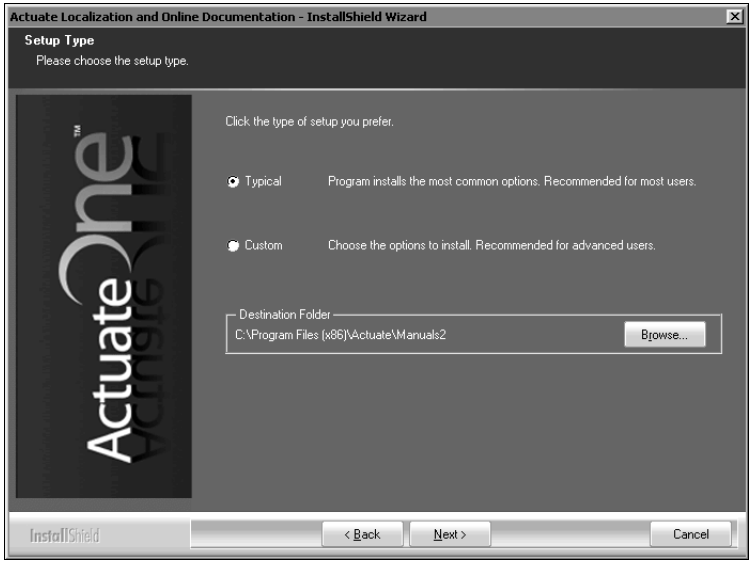


Figure 8-11 Specifying typical or custom setup type

4 In Start Copying Files, review the settings shown in Figure 8-12. Choose Next.

Figure 8-12 Reviewing settings before copying files

Setup Status displays an indicator showing how the installation is progressing, as shown in Figure 8-13.

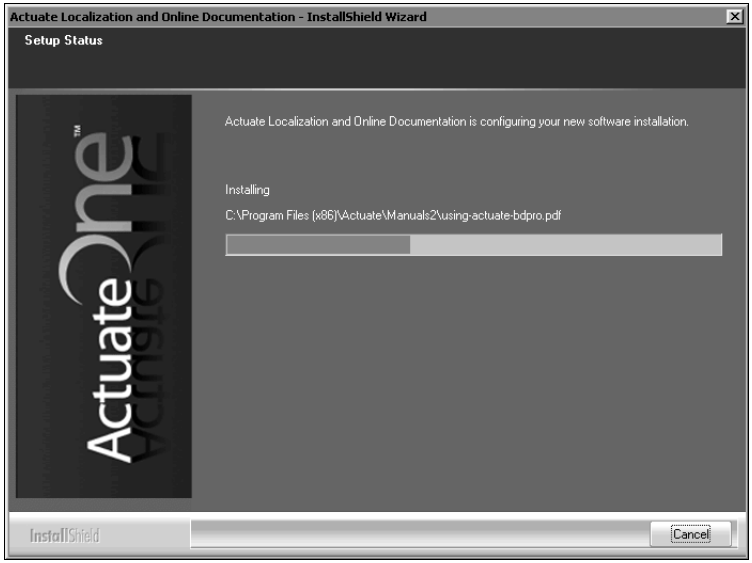


Figure 8-13 Viewing Setup Status

Setup completed successfully message appears, as shown in Figure 8-14. Choose OK.

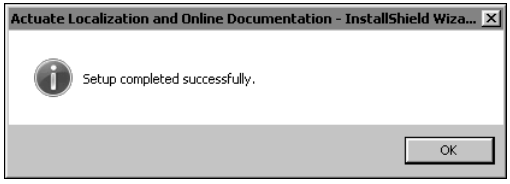


Figure 8-14 Viewing successful setup message

About accessing online help

iHub supports accessing online help in following ways:

- Online from www.actuate.com
  Use this option to ensure that you always have the latest documentation.
- Locally from the installed online localization and documentation files
  Use this option if you do not have an internet connection.

How to switch between online help and local help

1 Choose Start➤Programs➤Actuate➤Switch Help Location.

2 On Switch Help Location, select either Use online help, or Use local help, as shown in Figure 8-15.

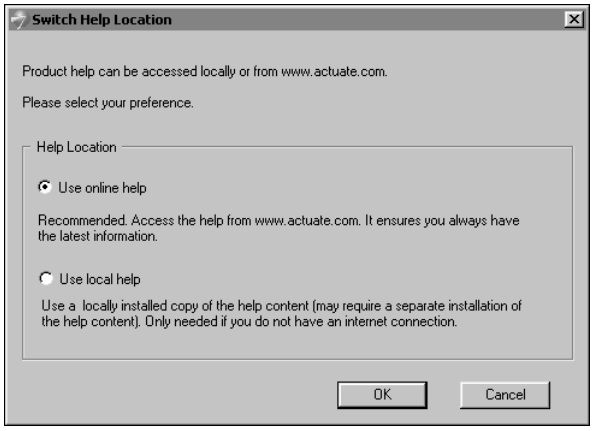


Figure 8-15 Selecting online or local help

Choose OK.

3 Restart Actuate BIRT iHub Service.

Searching PDF manuals using master-index.pdx

If you install the PDF version of the manuals, you can also use the Actuate Documents Catalog (master-index.pdx) to search for topics across the entire set of books.

How to search the Actuate Documents Catalog

1 Navigate to the \Program Files (x86)\Actuate\Manuals2 directory. Open master-index.pdx.

2 On Search, in Where would you like to search?, select All PDF documents in, then choose the \Program Files (x86)\Actuate\Manuals2.

3 In What word or phrase would you like to search for?, enter the word or phrase. For example, type accessing online help, as shown in Figure 8-16.

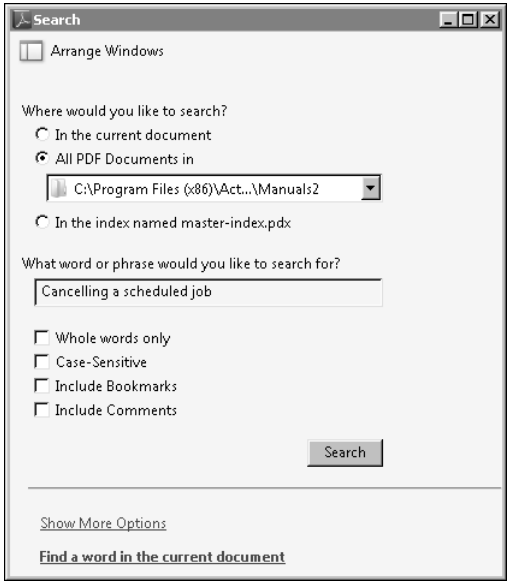


Figure 8-16 Specifying the search

Choose Search.

Search displays all occurrences of the word or phrase in the Actuate Documents Catalog.

4 Select an item in the results list to display the documentation in Adobe Reader, as shown in Figure 8-17.

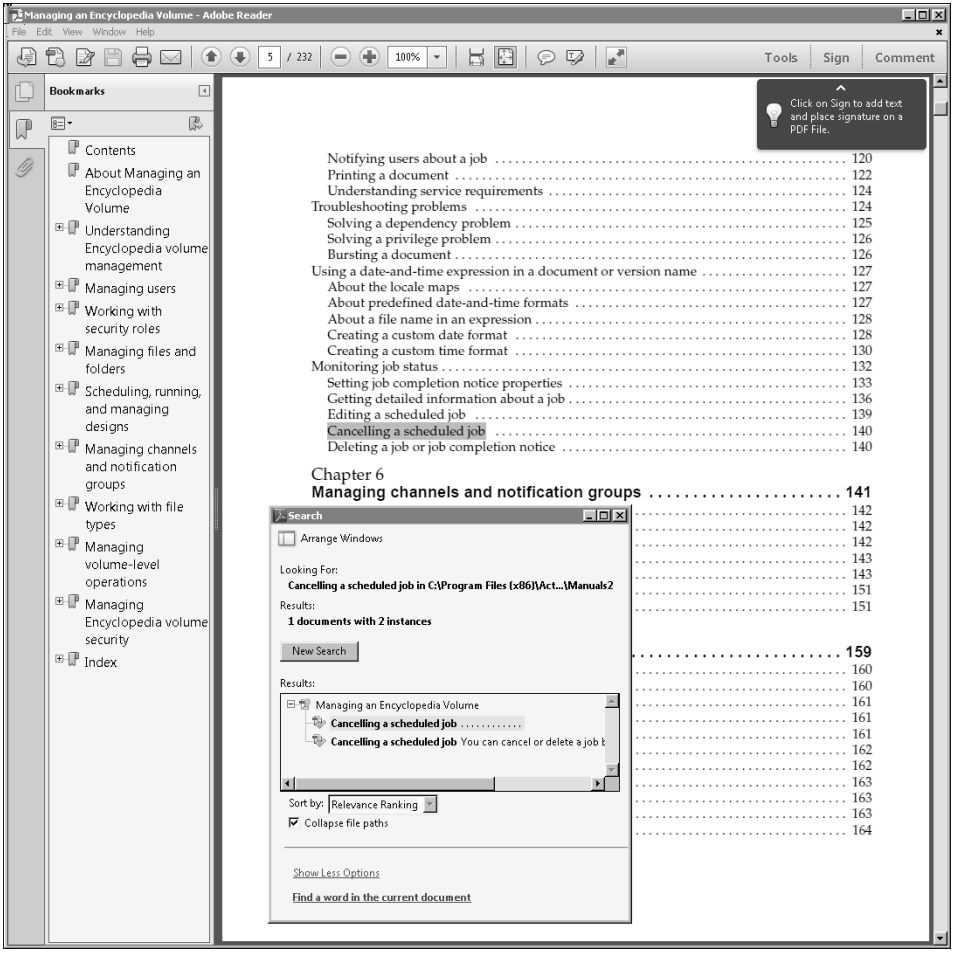


Figure 8-17 Viewing search results

Part Three

Licensing

Chapter

9

Licensing BIRT iHub

This chapter discusses the following topics:

- Understanding licensing types
- Understanding licensing options
- Installing Actuate BIRT iHub System license files
- Understanding CPU binding

Understanding licensing types

BIRT iHub System licensing supports running BIRT iHub with sets of features grouped as license options. You enable BIRT iHub System options using one or more of the following types of license models:

- Named User

  Specifies the maximum number of named users that can use a BIRT iHub System. A named user is a distinct individual who receives content and value from BIRT iHub.

  ABIRT iHub administrator must specify the options that a user can access in an Encyclopedia volume. The administrator makes this specification by configuring the user Licensed Options properties in Management Console.

  When you license an option, such as BIRT, BIRT 360, or BIRT Page Level Security Option, the license entitles the user to access a single volume in BIRT iHub System. If the user needs additional Encyclopedia volumes for multiple applications, archiving, or other purposes, you must license the Multi-Tenant Option for each additional Encyclopedia volume the user needs to access.

- CPU Core

  Specifies the maximum number of CPUs that BIRT iHub System can use. Any number of users can access the licensed options on the system provided adequate licensing and capacity exists.

- Instance

  A BIRT on Demand licensing option that provides a pre-packaged amount of dedicated capacity for a customer application. With instance licensing, the customer does not need to count named users. Multiple instances can be combined to meet capacity needs.

- Subscription

  An annual payment option that permits the use of the licensed software that includes maintenance. Offered with some of the other licensing models.

  A subscription license is not a perpetual license. Once the subscription term expires, the software cannot be used.

- Software as a Service (SaaS)

  Some products are offered as a Software as a Service (SaaS) option, providing customers with a solution without incurring the acquisition and management costs of hardware and traditional licenses.

- Packages

  Some options are offered as packages to customers for convenience and cost-saving benefits. These packages can be used in conjunction with individually selected options.

- Work Unit (WU) License

  Specifies iHub features and functionality using an aggregate model. This plan defines each iHub System resource as a work unit.

  Similar to CPU Core licensing, but defined at a more granular level. With Work Unit Licensing, the customer can license just the precise amount of capacity needed for application requirements. Any number of users can access the licensed options provided sufficient capacity has been purchased.

In a CPU Core and Work Unit licensing, Actuate currently uses the Standard Performance Evaluation Corporation (SPEC) standard benchmark for measuring machine capacity based on CPU, memory, disk, and network capacity.

Understanding licensing options

Table 9-1 lists and describes BIRT iHub System license options. BIRT iHub System options are separately licensed products. Some license options require other options to be licensed before their functionality is available to users. Table 9-1 also describes these prerequisites.

Table 9-1 BIRT iHub System license options

Option	Description	Supported releases
BIRT	Allows a user to publish and run a BIRT design using BIRT iHub. This option is a requirement for BIRT Page Level Security Option.	10, 11, iHub
BIRT 360	Allows a user to create, execute, and view dashboard files.	11, iHub
BIRT Data Analyzer	Allows a user to create, view, and modify cubeview files.	11, iHub
BIRT Interactive Viewer	Allows a user who has the BIRT Option to use BIRT Interactive Viewer to view and interact with a BIRT document.	10, 11, iHub
BIRT Page Level Security	Controls access to structured content available on the web. This option works for designs created using BIRT Designer Professional and requires the BIRT Option. Access privileges are based on user name or security role.	10, 11, iHub
e.Report Data Connector	Allows a BIRT design to access data that an Actuate Report Document (.roi) file contains. This option works for designs created using BIRT Designer Professional and requires the BIRT Option. Access privileges are based on user name or security role.	11, iHub

(continues)

Table 9-1 BIRT iHub System license options (continued)

Option	Description	Supported releases
Multi-Tenant	Allows a BIRT iHub System user to access more than one Encyclopedia volume. This option is available with a Unlimited User CPU License.	10, 11, iHub

To determine the license options installed on iHub, log in to Configuration Console, and choose Show License. The license options appear, as shown in Figure 9-1.

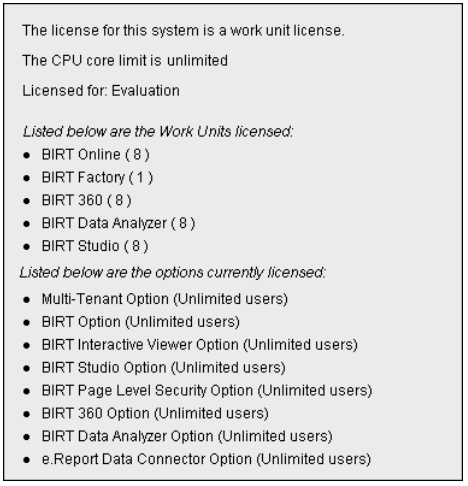


Figure 9-1 iHub License options

Installing Actuate BIRT iHub System license files

Actuate provides a license file to use when installing Actuate BIRT iHub System. New customers receive an e-mail containing a temporary BIRT iHub license file to use for the initial installation after Actuate processes the order. The temporary BIRT iHub System license expires 45 days after installation.

Actuate license enforcement for BIRT iHub requires a single, shared license for all nodes in a cluster. A design or document run using a temporary license appears with a watermark when viewed.

The name for the BIRT iHub license file uses the following format:

```
Actuate_iHub_key_xxxxxxx.xml
```

XXXXXXX is a unique seven-digit number generated by Actuate Licensing when it creates the license file.

Actuate BIRT iHub System customers perform an initial installation using a temporary license. After installing BIRT iHub System using the temporary license, the login screen displays two messages.

The following message about expiration of the initial license always appears on the login screen regardless of the license status:

```
Reminder
   Your BIRT iHub license expires in [the number of days] days, on
   [the specified date]. When the current license expires, the
   iHub will shut down and require a new license to restart.
   Please contact Actuate to purchase a new license.
```

The following message about how to obtain the second license file from Actuate Licensing appears until you install the new license issued by Actuate Licensing:

```
Reminder
   One or more iHubs in your BIRT iHub System are in violation of
   the node locked BIRT iHub license. After the grace period
   expires, the iHubs that violate the node locked BIRT iHub
   license cannot be restarted. Please contact Actuate Licensing
   (licensing@actuate.com or http://www.actuate.com/licensing), or
   your representative, and request a new license file for the
   iHub nodes that are in violation. Please restart the iHubs on
   the nodes after updating the license key file.
```

You have 45 days to apply for and install the license file after you install Actuate BIRT iHub System.

After installing BIRT iHub System, the installation informs a customer requiring a license to obtain the machine ID information on which Actuate BIRT iHub is running and transmit this information to Actuate Licensing. The machine ID is displayed in the reminder message. You can also use the utility, acmachineid, to obtain the machine ID. For information on how to use the acmachineid utility, see "How to use the acmachineid utility," later in this chapter.

After receiving the machine ID information, Actuate Licensing issues a new Actuate BIRT iHub System license file.

About the license file

This license file specifies the available iHub license options and node-key information for the cluster nodes. This license file must be in a shared location, specified by the <AC_CONFIG_HOME> attribute of the <Server> element in the acpmdconfig.xml file of each node, and accessible to all nodes in the cluster.

A node key associates an iHub node in a cluster with the machine ID. The node-key licensing mechanism restricts the iHub node installation to that machine.

On startup, each node in the cluster checks the shared license file, verifies the installed options, and determines whether its node key, which is generated at run time, matches the license information. If the node key matches, the node joins the cluster. Otherwise, it shuts down with an error if the node-lock-violation grace period has been exceeded.

A license file remains valid until a specific date. If your license file is about to expire, the system reminds you that the file expires on a certain date when you log in to the Configuration or Management Consoles. Reminders also appear in the system log file. To arrange for a permanent license file, or if you have a problem with an expiring file, please contact Actuate Licensing at licensing@actuate.com.

When upgrading a cluster node or installing iHub on a new machine, the customer must request a new license and supply the machine ID of the new machine.

Collecting machine information for a license

After installing BIRT iHub System using a temporary license file, such as an evaluation license, you must collect information about the machines running Actuate BIRT iHub software and send it to Actuate Licensing. During the installation process, the InstallShield Wizard prompts you to provide the location of the license file. After providing the location of the license file, the InstallShield wizard issues a prompt similar to the following message.:

```
The iHub system license file is locked to the machines that are
   used in the iHub system. The following machine id must be used
   to request a node key license file from Actuate:

   IORRHEHs6S5UCsEtrdVu6jOixmzvFY3BbOqXLiwswQGDceJmKYYaEu0j18lQxjM
   sYCxnka3hVkDZFGwkmQMxb+hgKaz4om2vLUcS0ocYTA7Ta6VTMavLFQo7bEjRyr
   olwxAKu0Vr4NA6o8uWCzjGZXX8KrjViSUoROj70hWOY=

Please contact Actuate Licensing (licensing@actuate.com or
   http://www.actuate.com/licensing), or your representative, and
   request a node locked iHub system license.

The machine id required for the node locked iHub system license
   can also be generated by using the acmachineid utility that can
   be found in the ACTUATE_HOME\iHub\bin folder.
```

The format of the alphanumeric string for the machine ID and location of the license file are different depending on the operating system. On a Windows system, the unique identifier for the network card is the source of the machine ID. You must have the network card enabled on the BIRT iHub machine to obtain the machine ID.

After installing iHub, you must run the utility, acmachineid, from the command line to generate the machine ID information. Copy the machine ID in the command prompt to a file or e-mail message and send it to Actuate Licensing. Actuate Licensing processes your request and sends the new license file for BIRT iHub System.

How to use the acmachineid utility

Use the acmachineid utility to obtain the machine ID information by performing the following tasks:

1. Open a command prompt and navigate to AC_SERVER_HOME\bin.
2. Type the following command and press Enter:

   ```
   acmachineid
   ```

   The utility provides output in the following format:

   ```
   C:\Program Files (x86)\Actuate\iHub2\bin>acmachineid
   STATUS:          OK
   GEN_VERSION:     22 iHub
   GEN_BUILD:       XxXBuild NumberXxX
   SERVERNAME:      <hostname>
   MACHINEID:
                    I0RREHs0Jk6tu0o8AbCrVL61x7kDpLlQKwS2t1W7qM67GbO8
                    VjcFs6pcuAgbtDaZauSbFFa2mRejwVJc7ZjKfMEVl1suXglM
                    KmZLiwtLykwJisqMS0EhYe5sCYoKjG+XL2UEnL2GGhLtI9f
                    JUMYzZORKk23jrxaSwUDsgKsvlc1A6q8UbmrrAYHD8Ggtpui
                    AmxWt4xjEM6rqlmsNEW/4ViMC0KDBkSn
   ```

   Figure 9-2 shows the output as it appears in the command prompt.

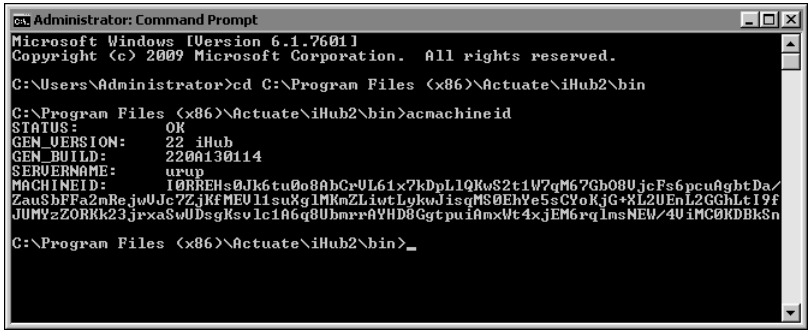


Figure 9-2 Executing acmachineid

Send Actuate Licensing the output of the acmachineid utility.

How to obtain a license file

To obtain a new license file for a licensed product or receive assistance with a license already issued to you, perform the following tasks:

1. Using a browser, go to the Actuate Support web site at the following location:

   `http://support.actuate.com`

2. Choose Downloads/Requests➛License Key Request.
3. Enter the required contact details and license key request information.
4. Choose Submit.

A maintenance customer should have login information for the Actuate e.Support web site. If you do not have access, please contact Actuate Support at support@actuate.com. You can also contact Actuate Customer Care at customercare@actuate.com.

If you are not a direct Actuate customer, contact the partner or distributor who supplies the product for the license file. If you have a problem obtaining a license file from this source, please contact Actuate Licensing at licensing@actuate.com.

Updating the Actuate BIRT iHub System license file

After performing an installation of Actuate BIRT iHub System and transmitting the required machine ID information to obtain a license, Actuate sends an e-mail containing an attached .txt (TXT) file. Replace the .txt extension with a .zip (ZIP) extension and open the file. This ZIP file contains the following files:

- readme.txt
  Instructions for installing Actuate BIRT iHub System using a license file and for obtaining a license file
- Actuate_iHub_key_XXXXXXX.xml
  Actuate BIRT iHub System license

An Actuate license file is an XML file. Actuate Licensing sends this XML file inside of a TXT file because transmitting a file with an .xml extension can cause problems in an e-mail system.

How to install the license file

To install the license file, perform the following steps:

1. Extract the contents of the ZIP file to a location on your local file system.
2. Log in to Configuration Console. For example, type http://localhost:8900 /acadmin/config/ in Address of a browser, and use the system configuration password that you specified during installation.
3. Choose Update License. If you do not see Update License, choose Simple view in the upper right corner.

4 On Actuate iHub update license, choose Browse to navigate to the location where you extracted the contents of the ZIP file. Select the Actuate BIRT iHub System license file and choose OK to apply the license.

If iHub requires a system restart to update the license file, the following message appears:

```
The license file cannot be applied without a server restart.
   Please copy the license file to the iHub license file
   location and restart the iHub system.
```

If this message appears, perform the following tasks:

1 Stop iHub system by choosing Stop system.

2 Copy the new license file to the shared location specified by the <AC_CONFIG_HOME> attribute of the <Server> element in the acpmdconfig.xml file. The <AC_CONFIG_HOME> attribute in the acpmdconfig.xml files for all nodes in a cluster point to this shared location.

3 Delete the old acserverlicense.xml file.

4 Rename the new license file to acserverlicense.xml.

5 Start iHub System.

5 Restart any node where the node-key configuration changed.

If you change the machine on which you installed Actuate BIRT iHub, you must re-apply to Actuate Licensing for a new license file. If you replace the network card on some machines, such as a Windows system, you may have to obtain a new license file since the unique identifier for the network card may be the source of the machine ID. If you have a license file installed and a reminder message appears when logging into Actuate Management Console, contact Actuate Licensing and provide the current Actuate iHub System license file with the output from the machine ID utility.

Actuate_iHub_key_XXXXX.xml will contain the node key information for the stand-alone machine or all machines in a cluster. There is no separate node license file for each machine.

Listing 9-1 shows the node key information the license contains, obtained from the acmachineid output you submitted to Actuate Licensing.

Listing 9-1 Viewing license node key information

```
<NodeKeys>
   <NodeKey
      MachineId="E0RREHs0Jk6tu0o8AbCrVL61x7kDpLlQKwS2t1W7qM67GbO8
      VjcFs6pcuAgbtZauSbFFa2mRejwVJc7ZjKfMEVl1suXglMKmZLiwtLykDa/
      wJisqMS0EhYe5sCYoKjG+XL2UEnL2GGhLtI9fJUMYzZORKk23jrxaSwUDig
      Ksvlc1A6q8UbmrrAYHD8GgtpuiAmxWt4xjEM6rqlmsNEW/4Vjm40KxlkSv"
      ServerName="W7CLSTRNODE1"/>
```

```
    <NodeKey
        MachineId="I0RREHs0Jk6tu0o8AbCrVL61x7kDpLlQKwS2t1W7qM67GbO8
        VjcFs6pcuAgbtZauSbFFa2mRejwVJc7ZjKfMEVl1suXglMKmZLiwtLykDa/
        wJisqMS0EhYe5sCYoKjG+XL2UEnL2GGhLtI9fJUMYzZORKk23jrxaSwUDsg
        Ksvlc1A6q8UbmrrAYHD8GgtpuiAmxWt4xjEM6rqlmsNEW/4ViMC0KDBkSn"
        ServerName="W7CLSTRNODE2"/>
</NodeKeys>
```

About modifying a license

If you decide later to license additional iHub options, the existing license file becomes invalid. You must install a new license file.

Contact Actuate Licensing for the new license file. If you are an Actuate international customer, please be aware that the e-mail message sent to Actuate goes to Actuate headquarters, and we route your request to a team in the appropriate country.

Understanding CPU binding

BIRT iHub System supports CPU binding on a machine with an appropriate CPU-based license. CPU binding restricts a process or processes to run on a subset of CPUs. If you bind the BIRT iHub System to a subset of CPUs, only those CPUs count toward the total number of licensed CPUs. The CPU limit in the license file applies to all CPUs for all machines in the cluster. Depending on the operating system and specific system command, you can restrict other processes from running on the processor to which you bind a process.

You can bind BIRT iHub processes to a specific set of processors on a machine that runs a Windows or Linux operating system. The default configuration does not bind BIRT iHub to a set of processors. In the default configuration, all processors on a BIRT iHub machine count toward the maximum number of licensed CPUs.

To bind BIRT iHub to a set of processors, bind the Actuate Process Management Daemon (PMD) to the processors. The Actuate PMD starts all BIRT iHub processes. The processes inherit the binding from the Actuate PMD.

In a cluster, BIRT iHub counts only the processors on nodes that join the cluster and run the encycsrvr process. An encycsrvr process runs when a node is online. BIRT iHub counts the number of processors on a machine when the first encycsrvr process starts.

When deploying BIRT iHub on a machine with multi-threaded CPUs that use logical processors, the customer receives a license based on the number of physical processors in the system. To accommodate the use of logical processors, the customer receives a license key that specifies two or four times the number of physical processors.

This section contains information on the following topics:

- Binding a BIRT iHub to processors on a Windows machine
- Checking BIRT iHub bound processors
- Configuring e-mail for CPU license problems

Binding a BIRT iHub to processors on a Windows machine

You can perform the following types of CPU binding on Windows:

- Binding to specific CPUs
- Binding to multiple-core CPUs
- Binding an Actuate process to a processor

The following sections describe these features.

Binding to specific CPUs

On a multiple-CPU machine running the Windows operating system, the server operating system assigns an ID number to each processor. Windows Task Manager lists the IDs of the available processors. The numbering starts at 0.

How to bind BIRT iHub to a set of processors

To bind BIRT iHub to a set of processors, perform the following steps:

1. Choose Start➛Control Panel➛System.

   On System Properties, choose Advanced. Then select Environment Variables.

2. On Environment Variables, perform the following tasks:

   1. In System Variables, choose New.
   2. On New System Variable, perform the following tasks:

      1. in Variable name, type:

         ```
         AC_PMD_WINDOWS_CPUS
         ```

      2. In Variable value, specify the processors to which to bind BIRT iHub by typing a comma-separated list of integers. For example, to bind BIRT iHub to CPU 0, CPU 3, and CPU 4, type the following list:

         ```
         0,3,4
         ```

      New System Variable looks like Figure 9-3.

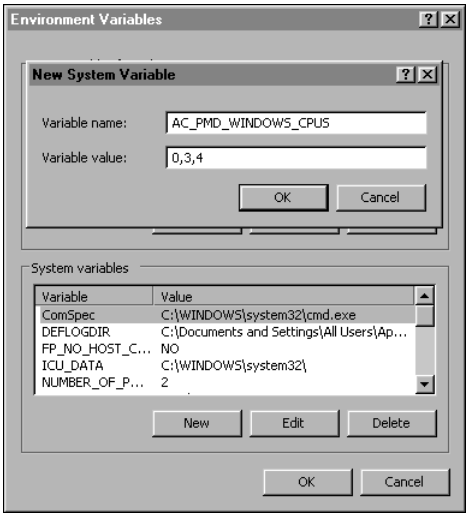


Figure 9-3 Creating the AC_PMD_WINDOWS_CPUS system variable

Choose OK. AC_PMD_WINDOWS_CPUS appears in System variables, as shown in Figure 9-4.

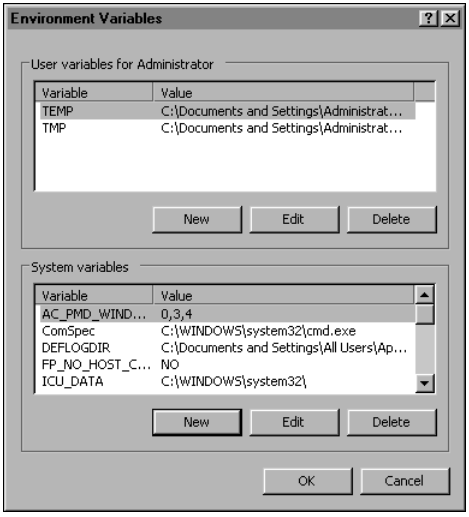


Figure 9-4 Viewing the AC_PMD_WINDOWS_CPUS system variable

On Environment Variables, choose OK. Then, on System Properties, choose OK.

You can verify the CPU binding by checking the Processor Affinity of the BIRT iHub process using Task Manager.

Binding to multiple-core CPUs

You can also perform multiple-core CPU binding, similar to the way you bind to a single CPU, using the AC_PMD_WINDOWS_CPUS environment variable, as described in the previous section. To BIRT iHub, each core appears as a logical CPU.

For example, on a dual-core, two-CPU system, setting the variable value to 0,1 binds BIRT iHub to both cores on the first CPU. Setting the variable value to 0,2 binds BIRT iHub to one core on each CPU. Setting the variable value to 0 binds BIRT iHub to one core on the first CPU.

Actuate does not recommend restricting BIRT iHub processing on a multiple-core CPU machine to one core for licensing purposes. BIRT iHub System achieves significant performance gains on a multiple-core CPU machine.

For example, BIRT iHub scales nearly perfectly from 1 to 2 cores and gets 50% better throughput on a dual-core system than on a two-CPU system.

Binding an Actuate process to a processor

If you bind the BIRT iHub PMD to a subset of CPUs on a machine, you can also bind the Factory, View, Integration, and Caching processes to a specific CPU. Under some conditions, binding an Actuate process to a specific CPU can enhance performance. Binding an Actuate process to a CPU has no affect on the CPU calculations BIRT iHub performs to determine the maximum number of licensed CPUs.

If you bind a process to a CPU, you must bind the CPU to both the BIRT iHub PMD and the process. BIRT iHub writes to the error log and stops the process if you bind a process to a CPU that you do not bind to the PMD.

To bind a BIRT iHub process to CPU processors, use the ProcessorAffinity element in the acserverconfig.xml file for BIRT iHub. List the IDs for the CPUs to which to bind a process as Item subelements in the following ProcessorAffinity elements:

- To bind Factory processes, specify the CPU IDs in the ProcessorAffinity element within the ReportingService element.
- To bind View processes, specify the CPU IDs in the ProcessorAffinity element within the ViewingService element.
- To bind Integration processes, specify the CPU IDs in the ProcessorAffinity element within the IntegrationService element.

You must also ensure that you bind the specified CPUs to the PMD for the BIRT iHub machine. For example, on a four-CPU machine, the following ProcessorAffinity example binds View processes to CPU IDs 0 and 2:

```
<ViewingService
     EnableViewingService="true"
   <ProcessorAffinity>
      <Item>0</Item>
      <Item>2</Item>
   </ProcessorAffinity>
/>
```

About processors and hyperthreading

Some Intel processors use hyperthreading, a technology that counts each physical processor as a specific number of logical processors. The operating system and any programs running on the machine see the number of logical processors, not the number of physical processors.

When a machine uses hyperthreading, Windows Task Manager lists the logical processors, not the physical ones. You specify the number of logical processors in the environment variable. When a machine uses hyperthreading, BIRT iHub calculates the number of bound processors by dividing the number of bound logical processors by the number of logical processors for each physical processor. If the result contains a decimal component, BIRT iHub uses the next highest integer. For example, it rounds 4.3 to 5. In the following example, a machine has four physical processors. With hyperthreading enabled, each physical processor corresponds to two logical processors. The machine has the following logical processors available:

- Physical processor 0 corresponds to logical processors 0 and 1.
- Physical processor 1 corresponds to logical processors 2 and 3.
- Physical processor 2 corresponds to logical processors 4 and 5.
- Physical processor 3 corresponds to logical processors 6 and 7.

If you bind BIRT iHub to the five logical processors 0, 2, 3, 6, and 7, it calculates the number of bound processors as:

```
5/2 = 2.5
```

BIRT iHub rounds this number up to determine that you have three bound processors.

Checking BIRT iHub bound processors

BIRT iHub performs the following bound processor checks:

- The number of processors a cluster uses
- The set of bound processors

Determining the number of processors an iHub System uses

When the PMD starts the first encycsrvr process on a machine, the PMD determines the number of processors to which BIRT iHub is bound and stores the list of bound processors.

If you change the processor binding, BIRT iHub does not recognize the changes until you shut down all encycsrvr processes on the machine and restart one of the encycsrvr processes.

For example, a cluster that has a maximum licensed CPU limit of nine processors consists of two nodes, machine A and machine B.

The machines have the following configuration:

- Machine A has four processors with no processor binding. All the processors can run Actuate processes. BIRT iHub manages an Encyclopedia volume.
- Machine B has eight processors with BIRT iHub bound to five processors. There is no encycsrvr process running on the machine, only the PMD.

The cluster counts four processors, the processors on machine A. If you start an encycsrvr process on machine B, BIRT iHub on machine A counts the five bound processors on the machine and increases the cluster processor count to nine, four on machine A and five on machine B.

If you bind the PMD on machine B to six processors, the change has no effect until you shut down all the running encycsrvr processes on machine B and restart an encycsrvr process on machine B.

After you stop the encycsrvr processes and restart an encycsrvr process on machine B, BIRT iHub System detects that the number of processors in the cluster is ten, which is greater than the maximum number of nine licensed processors. When the number of CPUs exceeds the number of CPUs your license permits, BIRT iHub does not start and returns an error message to Configuration Console.

Understanding CPU binding validation while iHub is running

When BIRT iHub is running, each encycsrvr process periodically compares the list of processors to which it is bound with the list to which it was bound when it started. If the lists differ:

- BIRT iHub writes a message with the processor information to the log file. The message contains the maximum number of processors the BIRT iHub license file permits and the following information:
  - Current and original number of bound processors
  - Current and original list of bound processors
- If configured, BIRT iHub sends an e-mail message to the administrator. The message states that the BIRT iHub System will shut down in one hour if the

list of bound processors is not corrected. The e-mail message contains the information that BIRT iHub sends to the log file.

You must rebind the encycsrvr process to the same processors to which it was originally bound.

During the next hour, any attempt to use the encycsrvr services fails and a message is written to the appropriate log file. If the list of processors is not restored after an hour, each BIRT iHub in the cluster shuts down and writes an error to its log file.

Understanding CPU binding validation when an Encyclopedia volume comes online

BIRT iHub uses a separate encycsrvr process to manage each Encyclopedia volume on a machine. When you take an Encyclopedia volume online, the PMD starts an encycsrvr process.

When the PMD starts an encycsrvr process, the PMD compares the list of processors to which the encycsrvr process is bound to the original list of processors to which the PMD is bound. If the lists differ:

- The encycsrvr process writes an error to its log file and shuts down.
- BIRT iHub does not take the volume online.

  A message in the configuration states that the binding of the new process differs from the original binding of the parent process.

Understanding CPU binding validation when running iHub processes

Each Factory and View process periodically compares its list of bound processors with the list of processors to which it was bound at startup. If the lists differ, the process writes an error to its log file and shuts down.

Configuring e-mail for CPU license problems

BIRT iHub System can send e-mail messages to an administrator if a change in processor binding violates the maximum number of licensed CPUs for BIRT iHub System. To send e-mail about a CPU license problem, set up BIRT iHub System by completing the following tasks in this order:

1. Configure every BIRT iHub node to send e-mail.
2. Specify the administrator e-mail address for BIRT iHub System.

   Specify an administrator e-mail address as the value for the Account to receive administrative e-mail parameter. Set the value by logging into Configuration Console, and choosing System—Properties—Advanced—Cluster Operation—Administrative.

For example, the following e-mail address sends e-mail to a user named admin at a company for which the domain is mycompany:

```
admin@mycompany.com
```

3. Restart BIRT iHub System. Restarting applies the changes after you set or change the e-mail address.

Part Four

Backing Up

Chapter

10

Backing up an Encyclopedia volume

This chapter discusses the following topics:

- Performing an Encyclopedia volume backup
- Backing up and restoring an Encyclopedia volume that uses a PostgreSQL database

Performing an Encyclopedia volume backup

When performing a backup, it is important to note that there are two types of data:

- Metadata

  Information about iHub system and Encyclopedia volume settings and data objects stored in third-party relational database management system (RDBMS) schemas.

- Data

  iHub system and Encyclopedia volume data objects, such as designs, documents, and information objects, stored as files on disk partitions, and the acserverconfig.xml file containing iHub configuration information.

The administrator must back up all Encyclopedia volume metadata and data to ensure the recoverability of a volume in the event of failure. In iHub, it is not necessary to back up the iHub system schema, although future versions may require this operation to protect critical system metadata. The administrator can restore a corrupted or missing system schema using the System Data Store Administrator utility. For more information on this utility, see "Specifying System Data Store Administrator properties" in Chapter 11, "Working with BIRT iHub utilities."

The third-party database that contains Actuate Encyclopedia metadata is a critical component of BIRT iHub System. An Actuate system administrator must take all necessary precautions to ensure that this database is properly backed up and available to safeguard Encyclopedia volume metadata. Please consult Actuate Support at the time of installation if you have any questions about the backup, recovery, or failover procedures necessary to protect against the possibility of catastrophic failure.

Managing the backup and recovery of Encyclopedia volume metadata and data files

A complete Encyclopedia volume backup must include the following items:

- A database backup of the Encyclopedia volume schema containing the metadata
- A copy of the folders from all Encyclopedia volume disk partitions containing file data
- A copy of the acserverconfig.xml file containing iHub configuration information

In the Windows BIRT iHub environment, the default AC_SERVER_HOME path is:

```
C:\Program Files (x86)\Actuate\iHub2
```

The default AC_DATA_HOME path is:

```
C:\Actuate\iHub\data
```

The default Encyclopedia volume path is:

```
C:\Actuate\iHub\data\encyc
```

The default acserverconfig.xml file path is:

```
C:\Actuate\iHub\data\config\iHub2
```

Back up the Encyclopedia volume metadata in the RDBMS at the same time that you back up the disk partition data files. A carefully coordinated backup ensures that a one-to-one correspondence exists between each entry in the volume metadata database and the data files.

The Encyclopedia volume metadata backup on the RDBMS must be done before the backup of the data on the disk partitions. Files that are partially created when the metadata backup begins are either not yet registered in the database or are marked incomplete in the database. The metadata database does not retain a record of incomplete files.

When contacting Actuate Support to troubleshoot problems, it is best to provide a snapshot of the Encyclopedia volume configuration, including the following items and information:

- A database backup of the Encyclopedia volume schema containing the metadata
- The name of the Encyclopedia volume schema and user that iHub uses to connect to the RDBMS
- A copy of the acserverconfig.xml file containing iHub configuration information
- A copy of the iHub logs

Using RDBMS and file system backup utilities

The administrator must perform the Encyclopedia volume metadata backup using the tools provided or supported by the RDBMS. Copying the physical files of a database at the operating system level while an RDBMS is running does not create a valid backup.

Most RDBMS backup tools can be scripted and run while iHub is using the database. PostgreSQL, Oracle, Microsoft SQL Server, and DB2 also provide graphical administration tools in addition to command-line tools. This chapter provides instructions on how to perform a backup in the PostgreSQL RDBMS

environment as a reference example. For more information on using other RDBMS systems and tools to back up and restore an Encyclopedia volume, see the vendor documentation.

How to perform an Encyclopedia volume backup

To back up an Encyclopedia volume, perform the following tasks:

1. Make sure that the autoarchive file purging process is not running.
2. Make an online backup of the volume schema using the tools provided by the RDBMS.
3. Back up the volume data files using the tools available in the operating system environment.

Avoiding conflict with the file purging process

A metadata backup is consistent with a data backup only if the file purging process that runs during an autoarchive operation does not occur between the time you back up the metadata and the time you back up the data. In Volumes—Properties—Advanced—Archiving And Purging, the administrator can specify when the file purging process runs.

How to configure Archiving And Purging

To configure the autoarchive file purging process, perform the following tasks:

1. From the advanced view of Configuration Console, choose Volumes.
2. On Volumes, point to the icon next to a volume name and choose Properties.

   In Volumes—Properties, choose Advanced. In Advanced, choose Archiving And Purging.
3. In Archiving And Purging, configure the following time-related file purging properties to times that do not conflict with the time when the backup operation runs, as shown in Figure 10-1:

   - Purge deleted files time
     Specifies the time when the file purging process runs to permanently delete expired files
   - Expiration time of deleted files
     Specifies the length of time that must elapse before the file purging process permanently deletes an expired file

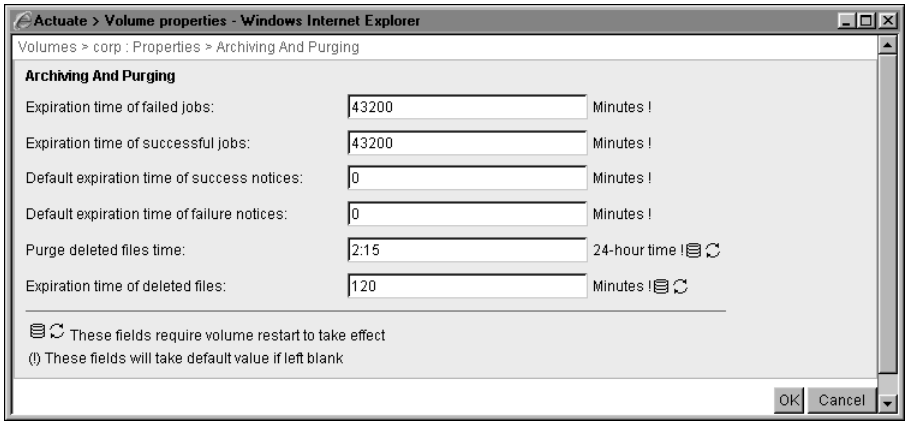


Figure 10-1 Configuring file purging properties

Choose OK.

For information on other aspects of archiving, see Chapter 12 "Archiving files," in *Configuring BIRT iHub*.

Backing up and restoring an Encyclopedia volume that uses a PostgreSQL database

PostgreSQL provides the pgAdmin graphical administration tool or the pg_dump and pg_restore command-line utilities to back up and restore a database. These PostgreSQL utilities run on the client not the server.

To back up an Encyclopedia volume in the OOTB PostgreSQL RDBMS environment, the administrator performs the following operations:

- Backs up Encyclopedia volume metadata using the pgAdmin graphical administration tool or the pg_dump PostgreSQL command-line utility
- Backs up Encyclopedia volume data and configuration files using operating system copy commands

Note that a backup of a PostgreSQL database is not portable across all operating systems.

To restore an Encyclopedia volume in the OOTB PostgreSQL RDBMS environment, the administrator performs the following operations:

- Restores Encyclopedia volume metadata using the pgAdmin graphical administration tool or the pg_restore PostgreSQL command-line utility
- Restores Encyclopedia volume data and configuration files using operating system copy commands

The following sections describe how to back up and restore an Encyclopedia volume that uses the OOTB PostgreSQL database to store the metadata. These demonstrations serve as a detailed reference example.

Backing up an Encyclopedia volume using pgAdmin

To back up an Encyclopedia volume using the pgAdmin graphical utility, perform the following tasks:

- Create a folder to contain the metadata and volume data backup files.
- Back up Encyclopedia volume metadata using the PostgreSQL pgAdmin utility.
- Back up the acserverconfig.xml file and volume data folders to the backup folder.

Create a folder to contain the metadata and volume data backup files outside the iHub data installation environment. To provide protection against single-point media failure, it is best to store the backup files on a partition that is physically separate from the Encyclopedia volume data location.

In a Windows environment, create a folder to contain the metadata and volume data backup files by performing the following tasks.

How to create a new backup folder

1. Open Windows Explorer.
2. In Windows Explorer, choose File➤New➤Folder repeatedly to create a set of nested folders in the following location:

   ```
   C:\Actuate\iHub\encyc_backup
   ```

Back up Encyclopedia volume metadata using the graphical administration tool, pgAdmin, to automatically run pg_dump by performing the following tasks.

How to run pg_dump using pgAdmin

1. In Windows, choose Start➤Programs➤pgAdmin III➤pgAdmin III.
2. On pgAdmin III, in Object browser, right-click the PostgreSQL Database Server and choose Connect, as shown in Figure 10-2. If the PostgreSQL Database Server does not appear in Object browser, you can add the server manually. For more information on adding the server manually, see Chapter 2, “Installing BIRT iHub,” earlier in this book.

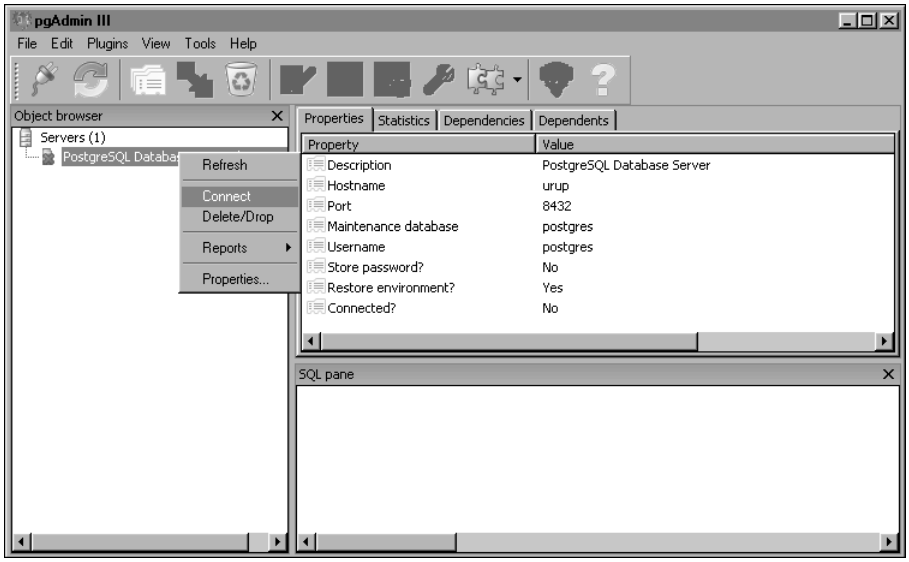


Figure 10-2 Connecting to PostgreSQL Database Server

3 In Connect to Server, type the postgres superuser password, as shown in Figure 10-3. You specified this password in PostgreSQL Database Information during the iHub installation.

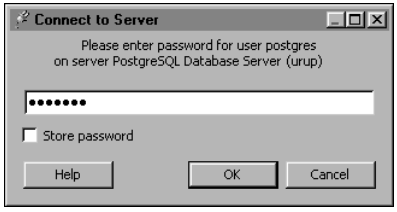


Figure 10-3 Typing the password to connect to PostgreSQL Database Server

4 On pgAdmin III, in Object browser, expand PostgreSQL Database Server, expand Databases, right-click iserver, and choose Backup, as shown in Figure 10-4. This operation backs up the entire iserver database. Alternatively, to back up only one Encyclopedia volume schema, such as ac_corp, right-click the volume, and choose Backup.

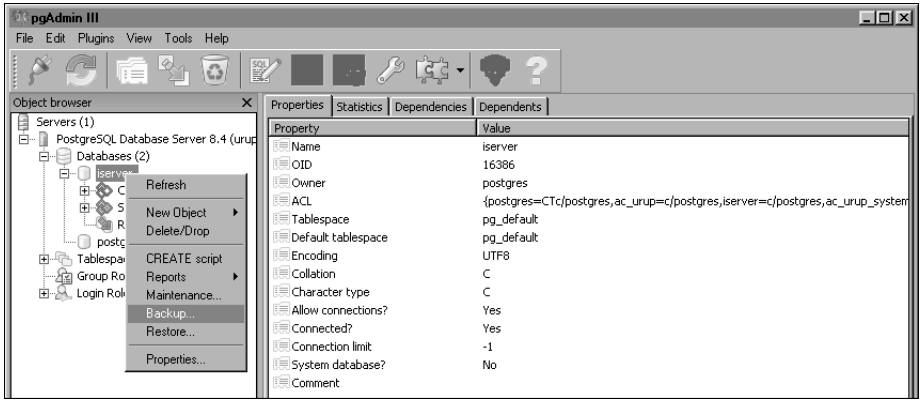


Figure 10-4 Choosing to back up the iserver database

5 On Backup Database iserver, perform the following tasks:

1 In Filename, type:

```
C:\Actuate\iHub\encyc_backup\iserver.backup
```

2 To execute pg_dump, accept the default option selections, as shown in Figure 10-5, and choose OK.

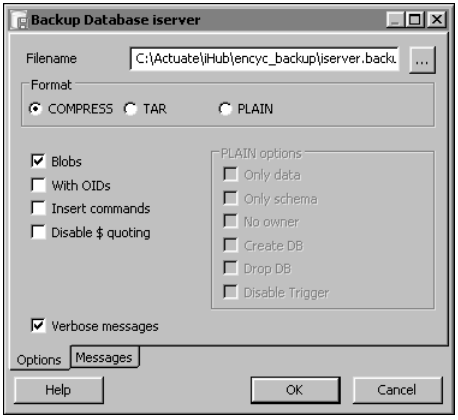


Figure 10-5 Choosing OK to start pg_dump execution

3 pg_dump executes, writing status messages to BackupDatabase iserver—Messages, as shown in Figure 10-6. Exit code 0 indicates that pg_dump ran successfully.

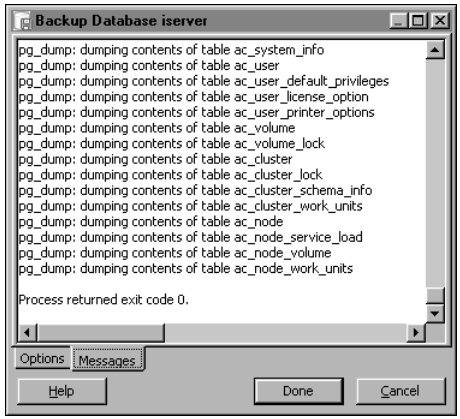


Figure 10-6 Viewing status messages pg_dump writes during execution

4 Scroll to the top of the output in Backup Database iserver—Messages to see the command that executed pg_dump, as shown in Figure 10-7.

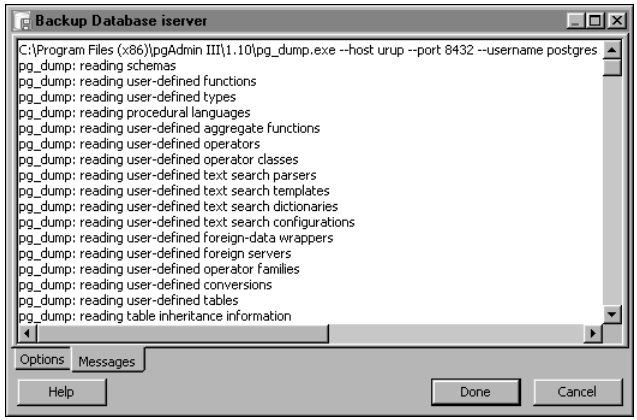


Figure 10-7 Viewing the command that executed pg_dump

The complete text of the command is:

```
C:\Program Files\pgAdmin III\1.10\pg_dump.exe --host urup
   --port 8432 --username postgres --format custom --blobs
   --verbose --file "C:\Actuate\iHub\encyc_backup\
   iserver.backup" iserver
```

5 On Backup Database iserver, choose Done.

Backing up an Encyclopedia volume using pg_dump

Alternatively, you can back up an Encyclopedia volume schema using the command-line version of pg_dump. The following example duplicates the operations performed in the previous section using the graphical PostgreSQL administration tool, pgAdmin. You do not need to do both activities.

The following example shows a typical pg_dump command used to export the contents of an Encyclopedia volume schema to a backup file:

```
pg_dump -F c -n ac_corp -f ac_corp_schema.dmp -h dbhost
   -p 8432 -U postgres dbname
```

This pg_dump command example uses the following arguments:

- F
  Specifies the output format. The value c is an abbreviation for custom, which creates a compressed archive that can be used as input to pg_restore.
- n
  Species the schema. Use multiple -n arguments to specify a list. Use wildcard notation to specify a character pattern, such as ac_*. to specify all volumes names that start with the prefix ac_. If -n is not specified, pg_dump exports all non-system schemas.
- f
  Specifies the output file, such as ac_corp_schema.dmp.
- h
  Specifies the host name of the machine where the PostgreSQL server is running, such as dbhost.
- p
  Specifies the port where the server listens for connection requests.
- U
  Specifies the user name for the connection to the PostgreSQL server, such as postgres.
- dbname
  Replace this string in the example with the database name, such as actuate_db.

Re-run the command to back up each Encyclopedia volume schema to a separate archive. To run multiple volume schema backups using a script, set up auto-login using a .pgpass file. The file should contain connection information in the following format:

```
hostname:port:database:username:password
```

More information about setting up a scripted backup using a .pgpass file is available at:

```
http://www.postgresql.org/docs/8.4/static/libpq-pgpass.html
```

Create a folder to contain the metadata and volume data backup files outside the iHub data installation environment. To provide protection against single-point media failure, it is best to store the backup files on a partition that is physically separate from the Encyclopedia volume data location.

Back up Encyclopedia volume metadata using pg_dump by performing the following tasks.

How to run pg_dump from a command prompt

1. Open a command prompt.
2. Navigate to the following location:

   ```
   C:\Program Files (x86)\Actuate\iHub2\postgresql\bin
   ```

3. Execute the following command. Substitute your machine name for urup in this example:

   ```
   pg_dump.exe --host urup --port 8432 --username postgres
      --format custom --blobs --verbose --file
      "C:\Actuate\iHub\encyc_backup\iserver.backup" iserver
   ```

   This operation backs up the entire iserver database. If the -n argument specifying a specific schema or list of schemas is not specified, pg_dump exports all non-system schemas. Alternatively, you can back up only one Encyclopedia volume schema, such as ac_corp, by using the -n argument to specify a particular schema.

4. Type the postgres superuser password. The administrator specified this password in PostgreSQL Database Information during the iHub installation procedure.

   pg_dump executes, writing status messages to the command prompt.

After backing up the Encyclopedia volume metadata, back up the acserverconfig.xml file and volume data directories to the backup directory by performing the following tasks.

How to back up the volume data folders

1. Open Windows Explorer and navigate to AC_DATA_HOME, which is the location of the iHub data. You specified this location on Setup Type during the install. The default path for AC_DATA_HOME is:

   ```
   C:\Actuate\iHub\data
   ```

2 In AC_DATA_HOME, navigate to the config folder that contains acserverconfig.xml file. In BIRT iHub, this file is located in the following subfolder:

```
C:\Actuate\iHub\data\config\iHub2
```

3 Select acserverconfig.xml, right click, and choose Copy.

Copy this file to the following backup location:

```
C:\Actuate\iHub\encyc_backup
```

4 In AC_DATA_HOME\encyc, select the file and filetype folders, and status folder, if it exists, right-click, and choose Copy. Copy these folders to the following backup location:

```
C:\Actuate\iHub\encyc_backup
```

In a backup taken immediately after an iHub installation, where there has been no activity on the system, the status folder may not exist. This folder contains information about job details and completion notices and does not appear until a job executes. If this folder is not present in the environment, simply back up the file and fileType folders.

Do not back up the postgresql folder in an Encyclopedia volume backup operation. The postgres folder contains data, such as log files, from the OOTB PostgreSQL RDBMS installation, which remains active. Inadvertently including these files in an iHub backup, then accidentally overwriting the files with a stale version in a restore operation can cause problems in the PostgreSQL RDBMS installation.

Restoring an Encyclopedia volume using pgAdmin

To restore a backed-up Encyclopedia volume, perform the following tasks:

- Take the Encyclopedia volume offline.
- Delete the acserverconfig.xml file and volume data folders in AC_DATA_HOME.
- Copy the backed-up acserverconfig.xml file and volume data folders from the backup folder to AC_DATA_HOME.
- Restore the Encyclopedia volume metadata using the PostgreSQL pg_restore utility.
- Take the Encyclopedia volume online.

To begin the restore operation, take the Encyclopedia volume offline by performing the following tasks.

How to take the Encyclopedia volume offline

1 Log in to Configuration Console. On the simple view, choose Advanced view. Choose Volumes.

2 On Volumes, take the volume offline, as shown in Figure 10-8.

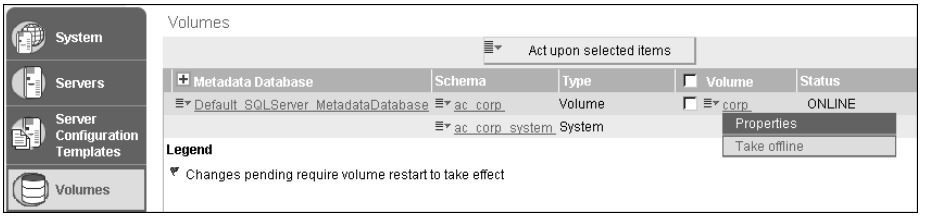


Figure 10-8 Taking the volume offline

How to restore the backed-up volume data folders

1 In Windows Explorer, navigate to AC_DATA_HOME\config\iHub2.

2 Select acserverconfig.xml, right-click, and choose Delete. Confirm the deletion.

3 In AC_DATA_HOME, open the encyc folder. In AC_DATA_HOME\encyc, select the file and filetype folders, and status folder, if it exists, right-click, then choose Delete. Confirm the deletion.

In a backup taken immediately after an iHub installation where there has been no activity on the system, the status folder may not exist. Be sure to not select and delete the postgresql folder.

4 In Windows Explorer, navigate to the following location:

```
C:\Actuate\iHub\encyc_backup
```

Select acserverconfig.xml, right-click, choose Copy, and copy this file to AC_DATA_HOME\config\iHub2.

5 In C:\Actuate\iHub\encyc_backup, select the file and filetype folders, and status folder, if it exists, right-click, choose Copy, and copy these folders to AC_DATA_HOME\encyc.

How to run pg_restore using pgAdmin

1 On pgAdmin III, in Object browser, right-click iserver and choose Restore, as shown in Figure 10-9.

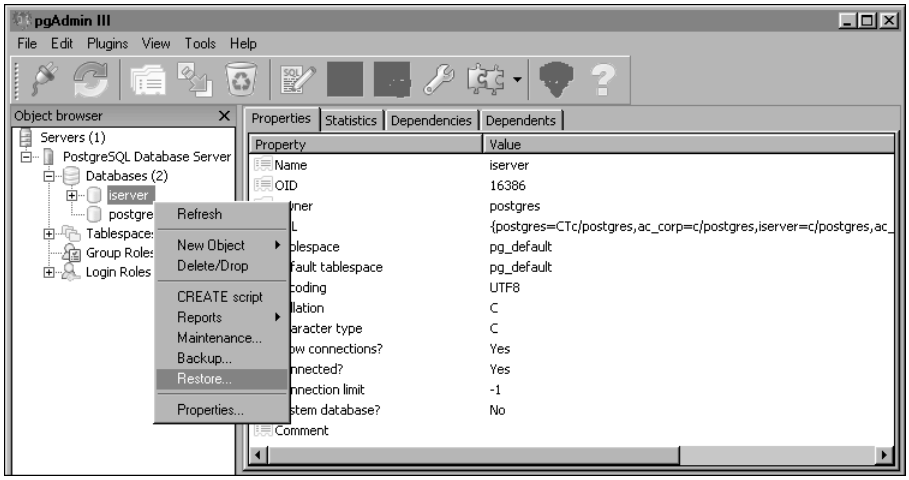


Figure 10-9 Choosing to restore the iserver database from backup

2 On Restore Database iserver, perform the following tasks:

1 Choose the ellipse next to Filename. On Select backup filename, navigate to, and select the backup file that pg_dump created. The name of this file is:

```
C:\Actuate\iHub\encyc_backup\iserver.backup
```

2 Select Clean before restore.

Restore Database iserver appears, as shown in Figure 10-10.

Figure 10-10 Specifying the backup file to restore

Choose OK.

3 pg_restore executes, writing status messages to BackupDatabase iserver—Messages, as shown in Figure 10-11. Exit code 0 indicates that pg_restore ran successfully.

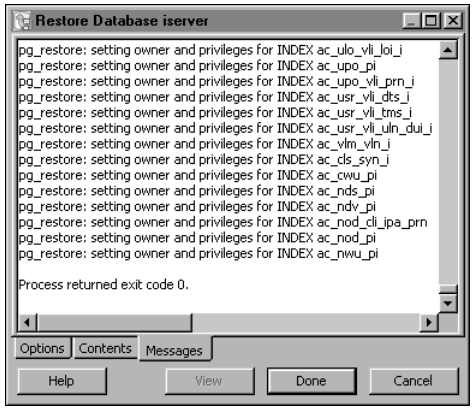


Figure 10-11 Viewing status messages pg_restore writes during execution

4 Scroll to the top of the output in Backup Database iserver—Messages to see the command that executed pg_restore, as shown in Figure 10-12.

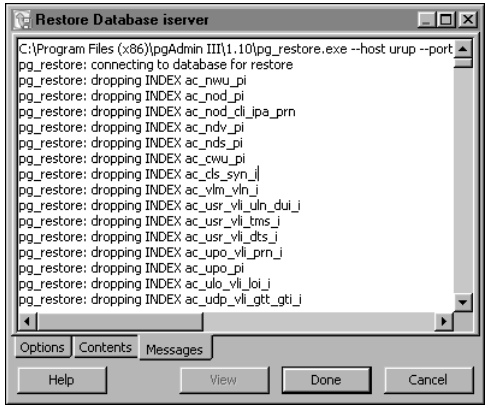


Figure 10-12 Viewing the command that executed pg_restore

The complete text of the command is:

```
C:\Program Files\pgAdmin III\1.10\pg_restore.exe --host urup
   --port 8432 --username postgres --dbname iserver --clean
   --verbose "C:\Actuate\iHub\encyc_backup\iserver.backup"
```

5 On Backup Database iserver, choose Done.

Restoring an Encyclopedia volume using pg_restore

Alternatively, you can restore an Encyclopedia volume schema using the command-line version of pg_restore. The following example duplicates the restore operations performed in the previous section using the graphical PostgreSQL administration tool, pgAdmin. You do not need to do both activities.

The pg_restore utility runs using arguments similar to the pg_dump utility. The following example shows a typical pg_restore command used to import the contents of a backup file to an Encyclopedia volume schema:

```
pg_restore -h mydbhost -p 8432 -U postgres -d db_name
   ac_corp_schema.dmp
```

Run pg_restore from the command line by performing the following tasks.

How to run pg_restore from a command prompt

1. Open a command prompt.
2. Navigate to the following location:

   ```
   C:\Program Files (x86)\Actuate\iHub2\postgresql\bin
   ```

3. Enter the following command. Substitute your machine name for urup in this example:

   ```
   pg_restore.exe --host urup --port 8432 --username postgres
      --dbname iserver --clean --verbose "C:\Actuate\iHub
      \encyc_backup\iserver.backup"
   ```

   Press Enter.

4. Type the postgres superuser password. The administrator specified this password in PostgreSQL Database Information during the iHub installation procedure. Press Enter.

   pg_restore executes, writing status messages to the command prompt.

Take the Encyclopedia volume online by performing the following tasks.

How to take the Encyclopedia volume online

1. Log in to Configuration Console. On the simple view, choose Advanced view. Choose Volumes.
2. On Volumes, take the volume online, as shown in Figure 10-13.

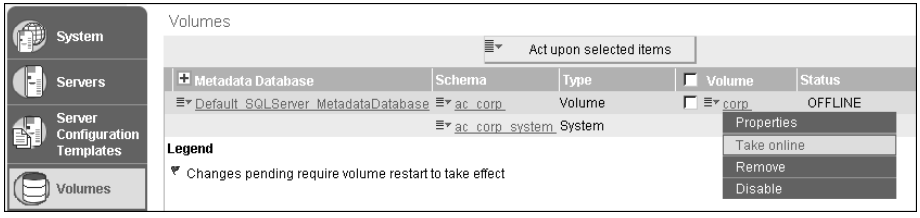


Figure 10-13 Taking the volume online

More information about backing up and restoring an Encyclopedia volume schema using the PostgreSQL pg_dump and pg_restore utilities is available at:

```
http://www.postgresql.org/docs/8.4/static/backup.html
```

Part Five

Utilities

Chapter

11

Working with BIRT iHub utilities

This chapter discusses the following topics:

- About BIRT iHub utilities
- Working with Encyclopedia Data Store Administrator
- Working with System Data Store Administrator
- Working with Encyclopedia Data Store Upgrader

About BIRT iHub utilities

The following sections provide information on how to specify properties and perform operations using the following utilities:

- Encyclopedia Data Store Administrator
- System Data Store Administrator
- Encyclopedia Data Store Upgrader

These utilities can be useful in performing administration tasks on iServer and iHub system and Encyclopedia schemas. When configuring Encyclopedia Data Store Administrator, System Data Store Administrator, or Encyclopedia Data Store Upgrader properties files, observe the following rules:

- Use / in path settings. \ is an escape character.
- Use Unicode property values. Save the property file in UTF-8 format, including the UTF-8 byte order mark (BOM).
- Use # at the start of a line to add a comment or selectively comment out properties.

When reading the property files, the utilities remove leading and trailing white space automatically, but preserve embedded white space.

Working with Encyclopedia Data Store Administrator

Use Encyclopedia Data Store Administrator to import or export volume metadata and create or delete a volume schema.

Specifying Encyclopedia Data Store Administrator properties

Table 11-1 describes the required Encyclopedia Data Store Administrator properties used to configure the environment for a manual import operation.

Table 11-1 Required Encyclopedia Data Store Administrator properties

Parameter	Description
AC_SERVER_HOME	Points to the location of the iHub binaries, specified during the BIRT iHub installation, as shown in Figure 4-4.

Table 11-1 Required Encyclopedia Data Store Administrator properties

Parameter	Description
DATABASE_TYPE	Type of supported RDBMS that contains the data store. Specify DB2, Oracle, PostgreSQL, or SQLServer.
LOG_FOLDER	Absolute path to the log folder.
SCHEMA_FILE_NAME	Base name of the file without the extension that contains the schema definition.
SCRIPT_HOME	Absolute path to the root of the folder hierarchy that contains the scripts and the schema definition.

Table 11-2 describes optional Encyclopedia Data Store Administrator properties. The properties used depend on the type of operation performed and the installation environment.

Table 11-2 Optional Encyclopedia Data Store Administrator properties

Parameter	Description	Default value	Supported databases
APPLICATION _USER	Application user ID for connecting to the database for normal operations. Must be a legal SQL identifier. Typically iserver. This parameter is required when creating or populating a schema. Restrict the iHub application user name to alphanumeric and underscore characters with an initial alphabetic character in the pattern [a-z][a-z 0-9]*. Do not use a hyphen.		All
APPLICATION _USER_PASSWORD	Application user password for connecting to the database for normal operations. Required when creating a schema. The password is not encrypted.		All

(continues)

Table 11-2 Optional Encyclopedia Data Store Administrator properties (continued)

Parameter	Description	Default value	Supported databases
CONFIG_SCHEMA _NAME	Specifies the schema definition in acserverconfig.xml. Use this property when any of the following properties is required but is not specified in the properties file: ■ APPLICATION_USER ■ DATABASE_HOST ■ DATABASE_INSTANCE ■ DATABASE_NAME ■ DATABASE_PORT ■ DATABASE_TYPE ■ ORACLE_TNS_NAMES_FILE ■ SCHEMA_NAME	False	All
CREATE_SCHEMA	Set to true to create a new schema, false to use an existing schema. Specifies whether to create a new Encyclopedia schema. PostgreSQL only, not supported for DB2, Microsoft SQL Server, or Oracle. Creating a schema automatically creates the schema owner and application user if necessary.	False	PostgreSQL
DATA_EXPORT _FOLDER	Absolute path to the folder to which data is exported. This parameter is required if EXPORT_DATA is true. Folder is on the database machine.		All
DATA_EXPORT _FORMAT	Format of exported data. Specify DB2, Oracle, PostgreSQL, or SQL Server.	{DATABASE _TYPE}	All
DATA_IMPORT _FOLDER	Absolute path to the folder from which data is imported. This parameter is required if IMPORT_DATA is true. For PostgreSQL, the data must be on the database server.		All
DATA_IMPORT _FORMAT	Format of imported data. Specify DB2, SQL Server, Oracle, or PostgreSQL.	{DATABASE _TYPE}	All

Table 11-2 Optional Encyclopedia Data Store Administrator properties (continued)

Parameter	Description	Default value	Supported databases
DATABASE_HOST	Hostname or IP address of the machine hosting the database. This value is required for PostgreSQL. Required for Oracle if not using TNS.		All
DATABASE _INSTANCE	Specifies the SQL Server database instance. The default value is the name of the database instance on the host machine and port. The default name of the SQL Server 2008 database instance is MSSQLSERVER.		SQL Server
DATABASE_NAME	Database name for PostgreSQL. For Oracle not using TNS, the database service name. For Oracle using TNS, the TNS net service name.		All
DATABASE_PORT	Port that the database server uses. This value is required for PostgreSQL. Required for Oracle if not using TNS.		All
DATABASE_TYPE	Type of RDBMS that contains the data store. iHub currently supports DB2, PostgreSQL, Microsoft SQL Server, and Oracle.		All
DEFAULT _DATABASE_NAME	Used by the superuser to connect to the default database in order to create the iHub application database. Required when the application database does not exist. PostgreSQL only.		PostgreSQL
DELETE_ALL _DATA	Set to true to delete all data from the data store. Does not delete data from other tables in the schema.	False	All
DELETE_DATA	Set to true to delete filtered data from the data store. This parameter only works when a filter value is available.	False	All
DEPOPULATE _SCHEMA	Set to true to delete data store elements such as tables, views, stored functions, and procedures from the schema. Does not remove any other objects in the schema.	False	All

(continues)

Table 11-2 Optional Encyclopedia Data Store Administrator properties (continued)

Parameter	Description	Default value	Supported databases
DROP_SCHEMA	Set to true to delete the schema.	False	PostgreSQL
EXPORT_ALL _DATA	Set to true to export all volumes from a schema.	False	All
EXPORT_DATA	Set to true to export a single volume from a schema. Use VOLUME_NAME to specify the name of the volume to export. Optionally, use NEW_VOLUME_NAME to specify a new name for the exported volume.	False	All
FILE_LOG_LEVEL	Minimum logging level for messages sent to the log file. This parameter only works for AcLogger, and is a standard JUL Level name. Supported values are CONFIG, FINE, FINER, and FINEST.	CONFIG	All
GENERATE _SCRIPTS	Set to true to generate scripts to perform operations instead of performing the operations directly	False	All
IMPORT_DATA	Set to true to import data into the data store.	False	PostgreSQL
INITIALIZE_DATA	Set to true to initialize the data in the data store, using the data initialization script.	False	All
LOG_FILE_COUNT	Maximum number of log files to create before starting to overwrite previous log files. Must be a valid integer. Data Store Administrator tools do not generate large logs.	5	All
LOG_FILE_NAME	Name of the log file. Do not add a file extension. The extension is set to .log. A unique number appends automatically to the file name to prevent overwriting previous logs.	Encyclopedia DataStore Administrator <number>.log	
LOG_FILE_SIZE	Approximate maximum size of log files, in byte, before a new log file starts. Must be a valid integer. Data Store Administrator tools do not generate large logs.	10,000,000	All

Table 11-2 Optional Encyclopedia Data Store Administrator properties (continued)

Parameter	Description	Default value	Supported databases
LOG_FOLDER	Full path of folder to write logs.	AC_DATA _HOME /server/log	
NEW_SCHEMA _NAME	Name of the new schema to be created, or the name of the target schema for data export. This name is required if {SCHEMA_NAME} is not present. Restrict schema name to alphanumeric and underscore characters with an initial alphabetic character in the pattern [a-z][a-z 0-9]*. Do not use a hyphen.	NEW _SCHEMA _NAME	All
NEW_VOLUME _NAME	Name of the new volume to be created, or the name of the target schema for data export. Do not use a hyphen in a volume name.	NEW _VOLUME _NAME	All
ORACLE_TNS _NAMES_FILE	Absolute path to the Oracle TNS names file. This path can be used instead of {DATABASE_HOST} and {DATABASE_PORT} to generate a JDBC URL.		Oracle
POPULATE _SCHEMA	Set to true to populate the schema with data store elements such as tables, views, stored functions, and procedures. Set to false if the data store elements already exist. Use Configuration Console to populate schema when possible. Schema automatically populates when defined in Configuration Console.	CREATE _SCHEMA	All
POPULATE _SCHEMA_PHASE _ONE	Set to true to create the schema objects necessary for data to be loaded. Set to false to prevent this schema object creation. Imports data without building indexes for fast load.	POPULATE _SCHEMA	All

(continues)

Table 11-2 Optional Encyclopedia Data Store Administrator properties (continued)

Parameter	Description	Default value	Supported databases
POPULATE _SCHEMA _ PHASE_TWO	Set to true to create schema objects phase one does not create. Set to false to prevent this schema object creation. Builds indexes and other ancillary structures in database.	POPULATE _SCHEMA	All
PROMPT_FOR _PASSWORDS	Indicates whether to prompt the user to type in a password instead of specifying it as a property value. Required if SCHEMA_PASSWORD is not present. Password prompting works only when running the tool from the command line.	True	All
SCHEMA_NAME	Name of the existing schema. Must be a legal SQL identifier. This parameter is required when performing operations on an existing schema. Restrict schema name to alphanumeric and underscore characters with an initial alphabetic character in the pattern [a-z][a-z 0-9]*. Do not use a hyphen.	SCHEMA _NAME	All
SCHEMA _PASSWORD	Password for the schema owner. Required if PROMPT_FOR _PASSWORDS is false.		All
SUPERUSER	User ID of the database superuser. This parameter is required if CREATE_SCHEMA is true or when importing data into PostgreSQL Typically postgres in a PostgreSQL database.		PostgreSQL
SUPERUSER _PASSWORD	Password of the database superuser. This parameter is required if CREATE_SCHEMA is true or when importing data into PostgreSQL.		PostgreSQL
TABLESPACE _NAME	Name of the application tablespace.		PostgreSQL
TABLESPACE _LOCATION	Absolute path to the application tablespace folder.		PostgreSQL
TIME_ZONE	Local time zone for installation.		All

Table 11-2 Optional Encyclopedia Data Store Administrator properties (continued)

Parameter	Description	Default value	Supported databases
VOLUME_NAME	Name of the volume or target schema for data import.	VOLUME _NAME	All

Performing operations using Encyclopedia Data Store Administrator utility

The Encyclopedia Data Store Administrator utility supports a wide range of export and import operations in the BIRT iHub environment. The following sections describe a mix of operational scenarios that an administrator can perform using this utility.

Importing one or more volumes into a new schema

This operation is only supported for PostgreSQL. NEW_SCHEMA_NAME does not need to match the schema name from which the utility exported the data. Do not change the volume names, since these names are defined in the exported data.

Creating a schema requires superuser privileges. The Encyclopedia Data Store Administrator utility creates the database and users if these items do not already exist.

Configure these properties as shown in the following example:

```
AC_SERVER_HOME = C:/Program Files (x86)/Actuate/iHub2
DATABASE_TYPE = PostgreSQL
DEFAULT_DATABASE_NAME = postgres
SUPERUSER = postgres
SUPERUSER_PASSWORD = <provide a password>
DATABASE_NAME = iserver
DATABASE_HOST = localhost
DATABASE_PORT = 8432
CREATE_SCHEMA = true
NEW_SCHEMA_NAME = <provide a name>
SCHEMA_PASSWORD = <provide a password>
APPLICATION_USER = iserver
APPLICATION_USER_PASSWORD = <provide a password>
IMPORT_DATA = true
DATA_IMPORT_FOLDER = {SQUIRREL_EXPORT_FOLDER}
```

Importing one or more volumes into a populated schema

This operation is only supported for PostgreSQL. Typically, you perform this operation to import an additional volume into a shared schema that already contains one or more volumes.

SCHEMA_NAME does not need to match the schema name from which the utility exported the data. Do not change the volume names, because these names are defined in the exported data. Importing data into PostgreSQL requires superuser privileges.

Configure these properties as shown in the following example:

```
AC_SERVER_HOME = C:/Program Files (x86)/Actuate/iHub2
DATABASE_TYPE = PostgreSQL
SUPERUSER = postgres
SUPERUSER_PASSWORD = <provide a password>
DATABASE_NAME = iserver
DATABASE_HOST = localhost
DATABASE_PORT = 8432
SCHEMA_NAME = <provide a name>
SCHEMA_PASSWORD = <provide a password>
IMPORT_DATA = true
DATA_IMPORT_FOLDER = {ISERVER OR IHUB_EXPORT_FOLDER}
```

Exporting all volumes from a schema

Configure these properties as shown in the following example:

```
AC_SERVER_HOME = C:/Program Files (x86)/Actuate/iHub2
DATABASE_TYPE = PostgreSQL
DATABASE_NAME = iserver
DATABASE_HOST = localhost
DATABASE_PORT = 8432
SCHEMA_NAME = <provide a name>
SCHEMA_PASSWORD = <provide a password>
EXPORT_ALL_DATA = true
NEW_SCHEMA_NAME = <provide a name>
DATA_EXPORT_FOLDER = C:/Projects/DataStores/Data
DATA_EXPORT_FORMAT = Oracle
```

You can omit NEW_SCHEMA_NAME if it is the same as SCHEMA_NAME.

Exporting a single volume from a schema

Configure these properties as shown in the following example:

```
AC_SERVER_HOME = C:/Program Files (x86)/Actuate/iHub2
DATABASE_TYPE = PostgreSQL
DATABASE_NAME = iserver
DATABASE_HOST = localhost
DATABASE_PORT = 8432
SCHEMA_NAME = <provide a name>
SCHEMA_PASSWORD = <provide a password>
EXPORT_DATA = true
VOLUME_NAME = <provide a name>
```

```
NEW_SCHEMA_NAME = <provide a name>
NEW_VOLUME_NAME = <provide a name>
DATA_EXPORT_FOLDER = D:/Projects/DataStores/Data
DATA_EXPORT_FORMAT = Oracle
```

You can omit NEW_SCHEMA_NAME if it is the same as SCHEMA_NAME. You can omit NEW_VOLUME_NAME if it is the same as VOLUME_NAME.

Deleting all volumes from a schema

Perform this operation to remove all volumes from the schema that SCHEMA_NAME specifies.

Configure these properties as shown in the following example:

```
AC_SERVER_HOME = C:/Program Files (x86)/Actuate/iHub2
DATABASE_TYPE = PostgreSQL
DATABASE_NAME = iserver
DATABASE_HOST = localhost
DATABASE_PORT = 8432
SCHEMA_NAME = <provide a name>
SCHEMA_PASSWORD = <provide a password>
DELETE_ALL_DATA = true
```

Deleting a single volume from a schema

Perform this operation to remove the volume that VOLUME_NAME specifies from the schema that SCHEMA_NAME specifies.

Configure these properties as shown in the following example:

```
AC_SERVER_HOME = C:/Program Files (x86)/Actuate/iHub2
DATABASE_TYPE = PostgreSQL
DATABASE_NAME = iserver
DATABASE_HOST = localhost
DATABASE_PORT = 8432
SCHEMA_NAME = <provide a name>
SCHEMA_PASSWORD = <provide a password>
DELETE_DATA = true
VOLUME_NAME = <provide a name>
```

Creating a new volume in an empty schema

iHub performs this operation when you create a new volume in Configuration Console and bring the volume online. When creating a new volume using Encyclopedia Data Store Administrator, set INITIALIZE_DATA to true and specify a name for NEW_VOLUME_NAME.

Setting POPULATE_SCHEMA to true to have Encyclopedia Data Store Administrator create schema elements, such as tables and indexes in an empty

schema. The schema owner and application user must already exist and have appropriate privileges.

Configure these properties as shown in the following example:

```
AC_SERVER_HOME = C:/Program Files (x86)/Actuate/iHub2
DATABASE_TYPE = Oracle
DATABASE_NAME = xe
DATABASE_HOST = localhost
DATABASE_PORT = 1521
SCHEMA_NAME = <provide a name>
SCHEMA_PASSWORD = <provide a password>
APPLICATION_USER = iserver
POPULATE_SCHEMA = true
INITIALIZE_DATA = true
NEW_VOLUME_NAME = <provide a name>
TIME_ZONE = America/Los_Angeles
```

Populating an empty schema

iHub performs this operation when you create a new schema in Configuration Console. Before populating an empty schema using Encyclopedia Data Store Administrator, create the schema owner and application user if they do not exist and give them the appropriate privileges.

```
AC_SERVER_HOME = C:/Program Files (x86)/Actuate/iHub2
DATABASE_TYPE = Oracle
DATABASE_NAME = xe
DATABASE_HOST = localhost
DATABASE_PORT = 1521
SCHEMA_NAME = <provide a name>
SCHEMA_PASSWORD = <provide a password>
APPLICATION_USER = iserver
POPULATE_SCHEMA = true
```

Creating a new volume in a populated schema

iHub performs this operation when you create a new volume in Configuration Console and bring the volume online. Typically, you perform this operation to add a volume to a shared schema that already contains one or more volumes.

When creating a new volume using Encyclopedia Data Store Administrator, set INITIALIZE_DATA to true and specify a name for NEW_VOLUME_NAME. The schema owner and application user must already exist and have appropriate privileges.

Configure these properties as shown in the following example:

```
AC_SERVER_HOME = C:/Program Files (x86)/Actuate/iHub2
DATABASE_TYPE = Oracle
```

```
DATABASE_NAME = xe
DATABASE_HOST = localhost
DATABASE_PORT = 1521
SCHEMA_NAME = <provide a name>
SCHEMA_PASSWORD = <provide a password>
APPLICATION_USER = iserver
INITIALIZE_DATA = true
NEW_VOLUME_NAME = <provide a name>
TIME_ZONE = America/Los_Angeles
```

Creating and initializing a new volume in a new schema

This operation is only supported for PostgreSQL. iHub typically performs this operation when you create a new schema and volume in Configuration Console and bring the volume online.

Creating a schema requires superuser privileges. Encyclopedia Data Store Administrator creates the database and users if these items do not already exist.

Configure these properties as shown in the following example:

```
AC_SERVER_HOME = C:/Program Files (x86)/Actuate/iHub2
DATABASE_TYPE = PostgreSQL
DATABASE_NAME = iserver
DATABASE_HOST = localhost
DATABASE_PORT = 8432
DEFAULT_DATABASE_NAME = postgres
SUPERUSER = postgres
SUPERUSER_PASSWORD = <provide a password>
CREATE_SCHEMA = true
NEW_SCHEMA_NAME = <provide a name>
SCHEMA_PASSWORD = <provide a password>
APPLICATION_USER = iserver
APPLICATION_USER_PASSWORD = <provide a password>
INITIALIZE_DATA = true
NEW_VOLUME_NAME = <provide a name>
TIME_ZONE = America/Los_Angeles
```

Working with System Data Store Administrator

Use the System Data Store Administrator utility to perform the following tasks:

- Create or delete a schema.
- Populate or depopulate a schema.

Specifying System Data Store Administrator properties

The system data store is a separate schema in the database that stores iHub system metadata. The system schema is a required element for any iHub installation. In a cluster, the nodes share system schema metadata and use this information for communicating and coordinating processing.

In BIRT iHub, it is not necessary to back up the system schema, although future versions may require this procedure to protect critical system metadata. If a system schema becomes corrupted or is accidentally deleted, the administrator can use the System Data Store Administrator utility to create a new schema. You can define a new system schema in Configuration Console or edit acserverconfig.xml to refer to the current system schema.

Before running the System Data Store Administrator utility, edit the PATH environment variable on your machine to contain the following string:

```
<AC_SERVER_HOME>\bin
```

where AC_SERVER_HOME refers to the iHub installation path. For example, using the default value for AC_SERVER_HOME, add:

```
C:\Program Files\Actuate\iHub\bin
```

To run the System Data Store Administrator utility, perform the following tasks:

1. In Windows Explorer, navigate to AC_SERVER_HOME\bin.
2. Create a properties file containing the property definitions necessary to perform an operation, such as creating and populating a system schema, described in "Performing operations using the System Data Store Administrator utility," later in this book. You pass this file to the System Data Store Administrator utility when you execute it.
3. Open a command prompt and navigate to AC_SERVER_HOME/bin.
4. Create a batch file or run the administrate_system_data_store.bat file, passing in the name of the properties files as an argument, using the following command line syntax:

   ```
   administrate_system_data_store systemdatastore.properties
   ```

   Alternatively, you can execute the utility by running the adminsds.bat script, using the following command-line syntax:

   ```
   adminsds systemdatastore.properties
   ```

   The batch file performs the following operations, as shown in Listing 11-1.

   - Checks to see if the administrator submitted a property file on the command line when running the script.

- If the administrator does not submit an argument, the script echoes a usage statement that describes the command-line syntax.
- Calls the set_tools_environment.bat script, which sets the environment variables.
- Executes the System Data Store Administrator utility using the name of the properties file as an argument.

Listing 11-1 administrate_system_data_store.bat

```
@echo off
if %1.==. goto :HELP

:: Set up environment variables
CALL "%~dp0"set_tools_environment.bat

:: Administrate data store
java com.actuate.iserver.system.datastore.admin
   .SystemDataStoreAdministrator %1

GOTO :END

:HELP
echo Usage: administrate_system_data_store.bat ^<properties
   file name^>

:END
```

In the example, the ^ symbol functions as a line-continuation marker to concatenate the lines together.

The SystemDataStoreAdministrator class has the same parent class as the Encyclopedia Data Store Administrator and uses the same property settings. For more information about these properties, see Table 11-1.

System Data Store Administrator properties include the following categories:

- Properties that specify details of the iHub installation environment, such as AC_SERVER_HOME and AC_DATA_HOME
- Database properties that specify the RDBMS type and JDBC connection details, such as the schema, application user, superuser, and passwords
- Schema operation properties that specify an action to perform, such as create, populate, or delete a schema
- Logging properties that control messages sent to the console and log files
- Engineering properties used by Actuate Support or Professional Services to assist with diagnosing or resolving specific issues

Refer to the required and optional properties tables in "Specifying Encyclopedia Data Store Administrator properties," earlier in this chapter, for detailed information about these properties.

Performing operations using the System Data Store Administrator utility

Creating and populating a new system data store schema is only supported for PostgreSQL. Creating and populating a schema requires superuser privileges. iHub performs this operation automatically when you create a new system schema in Configuration Console. The System Data Store Administrator utility creates the database and users if these items do not already exist.

Configure the properties as shown in the following example:

```
AC_SERVER_HOME = C:/Program Files (x86)/Actuate/iHub2
DATABASE_TYPE = PostgreSQL
DATABASE_NAME = iserver
DATABASE_HOST = localhost
DATABASE_PORT = 8432
DEFAULT_DATABASE_NAME = postgres
SUPERUSER = postgres
SUPERUSER_PASSWORD = <provide a password>
APPLICATION_USER = iserver
APPLICATION_USER_PASSWORD = <provide a password>
CREATE_SCHEMA = true
NEW_SCHEMA_NAME = <provide a name>
SCHEMA_PASSWORD = <provide a password>
INITIALIZE_DATA = true
```

Working with Encyclopedia Data Store Upgrader

Use Encyclopedia Data Store Upgrader to migrate an Encyclopedia volume manually from an earlier iServer Release 11 installation, such as Service Pack 3, to Release 11 Service Pack 4.

Table 11-3 describes the required Encyclopedia Data Store Upgrader properties to specify in the upgrade_encyclopedia_data_store.bat or other properties file for a manual upgrade operation.

Table 11-3 Required Encyclopedia Data Store Upgrader properties

Parameter	Description
AC_SERVER_HOME	Points to the location of the iHub binaries, which you specify during the BIRT iHub installation.

Table 11-3 Required Encyclopedia Data Store Upgrader properties

Parameter	Description
APPLICATION_USER	User ID used to connect to the database for normal operations.
DATABASE_TYPE	Type of relational database system that contains the data store. Actuate iHub currently supports PostgreSQL and Oracle.
LOG_FOLDER	Absolute path to the log folder.
SCHEMA_FILE_NAME	Base name of the file without the file extension that contains the meta-schema definition.
SCHEMA_NAME	Name of the target schema which the Encyclopedia Data Store Upgrader updates. Required if NEW_SCHEMA_NAME is not present. Restrict the schema name to alphanumeric and underscore characters with an initial alphabetic character in the pattern [a-z][a-z 0-9]*. Do not use a hyphen.
SCRIPT_HOME	Absolute path to the root of the folder hierarchy containing scripts and the meta-schema definition.

Table 11-4 describes the optional Encyclopedia Data Store Upgrader properties to specify in the upgrade_encyclopedia_data_store.bat or other properties file.

Table 11-4 Optional Encyclopedia Data Store Upgrader properties

Parameter	Description	Default value	Supported databases
CONFIG_SCHEMA _NAME	Specifies the schema definition in acserverconfig.xml. The schema name can be different from the database schema name.	False	All
DATABASE_HOST	Hostname or IP address of the machine hosting the database. This value is required for PostgreSQL. Required for Oracle if not using TNS.	False	All
DATABASE _INSTANCE	RDBMS instance that manages the database.		SQL Server

(continues)

Table 11-4 Optional Encyclopedia Data Store Upgrader properties (continued)

Parameter	Description	Default value	Supported databases
DATABASE_NAME	Database name for PostgreSQL. For Oracle not using TNS, the database service name. For Oracle using TNS, the TNS net service name.		All
DATABASE_PORT	Port that the database server uses. This value is required for PostgreSQL. Required for Oracle if not using TNS.		All
FILE_LOG_LEVEL	The minimum logging level for messages sent to the log file. This parameter only works for AcLogger. Supported values are CONFIG, FINE, FINER, and FINEST.	CONFIG	All
GENERATE_SCRIPTS	Set to true to generate scripts to perform operations instead of performing the operations directly.	False	All
LOG_FILE_COUNT	Maximum number of log files to create before starting to overwrite previous log files. Must be a valid integer. Encyclopedia Data Store Upgrader tool does not generate large logs.	5	All
LOG_FILE_NAME	Name of the log file. Do not add a file extension. The extension is set to .log. Standard JUL.FileHandler placeholders are supported. A unique number is appended automatically to the file name to prevent overwriting previous logs.		All
LOG_FILE_SIZE	Approximate maximum size of log files, in byte, before a new log file starts. Must be a valid integer. Encyclopedia Data Store Upgrader tool does not generate large logs.	10,000,000	All

Table 11-4 Optional Encyclopedia Data Store Upgrader properties (continued)

Parameter	Description	Default value	Supported databases
LOG_FOLDER	The full path of folder to write logs.	AC_DATA _HOME /server/log	All
ORACLE_TNS _NAMES_FILE	Absolute path of the Oracle TNS names file to use instead of DATABASE_HOST and DATABASE_PORT to generate a JDBC URL.		Oracle
PROMPT_FOR _PASSWORDS	Indicates whether to prompt the user to type in a password instead of specifying it as a property value. Required if SCHEMA_PASSWORD is not present. Password prompting works only when running the tool from the command line.	True	All
SCHEMA_PASSWORD	Password for the schema owner. Required if PROMPT_FOR _PASSWORDS is false.		All

Index

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Z