One Design
One Server
One User Experience

Installing BIRT iHub for Linux

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

www.actuate.com

Document No. 130131-2-430343 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 BIRT iHub 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

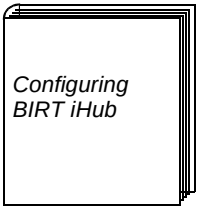
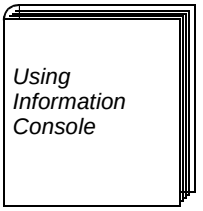
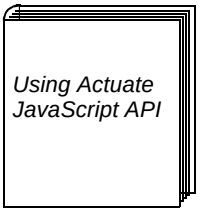
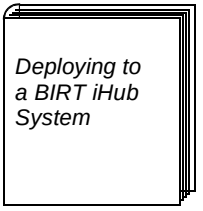
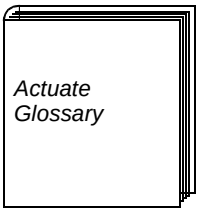
For information about this topic	See the following resource
Installing BIRT iHub for Linux	 *Installing BIRT iHub for Linux*
Installing BIRT iHub for Windows	 *Installing BIRT iHub for Windows*
Late-breaking information and documentation updates	Release notes and updated localization, HTML help, and 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	 *Configuring BIRT iHub*

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	*Managing an Encyclopedia Volume*
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	*Information Console Developer Guide*
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	*Using BIRT iHub Integration Technology*

(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 designs	
Using Actuate JavaScript API ■ Overview of programming with Actuate JavaScript ■ Creating custom pages using Actuate JavaScript ■ Reference for BIRT 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 /home/actuate/AcServer/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 /home/actuate/AcServer/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.rptdocument
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 Linux*

Installing BIRT iHub for Linux 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 Linux 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 Linux environment.
- *Chapter 4. Migrating to BIRT iHub.* Describes how to migrate to BIRT iHub in a Linux environment.

- *Chapter 5. Installing a BIRT iHub cluster.* Describes how to install a BIRT iHub cluster node in Linux.
- *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 Linux.
- *Chapter 8. Installing iHub Integration Technology and documentation.* Describes how to install BIRT iHub Integration Technology and Documentation in Linux.
- *Part 3. Licensing.* Describes the licensing for 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.

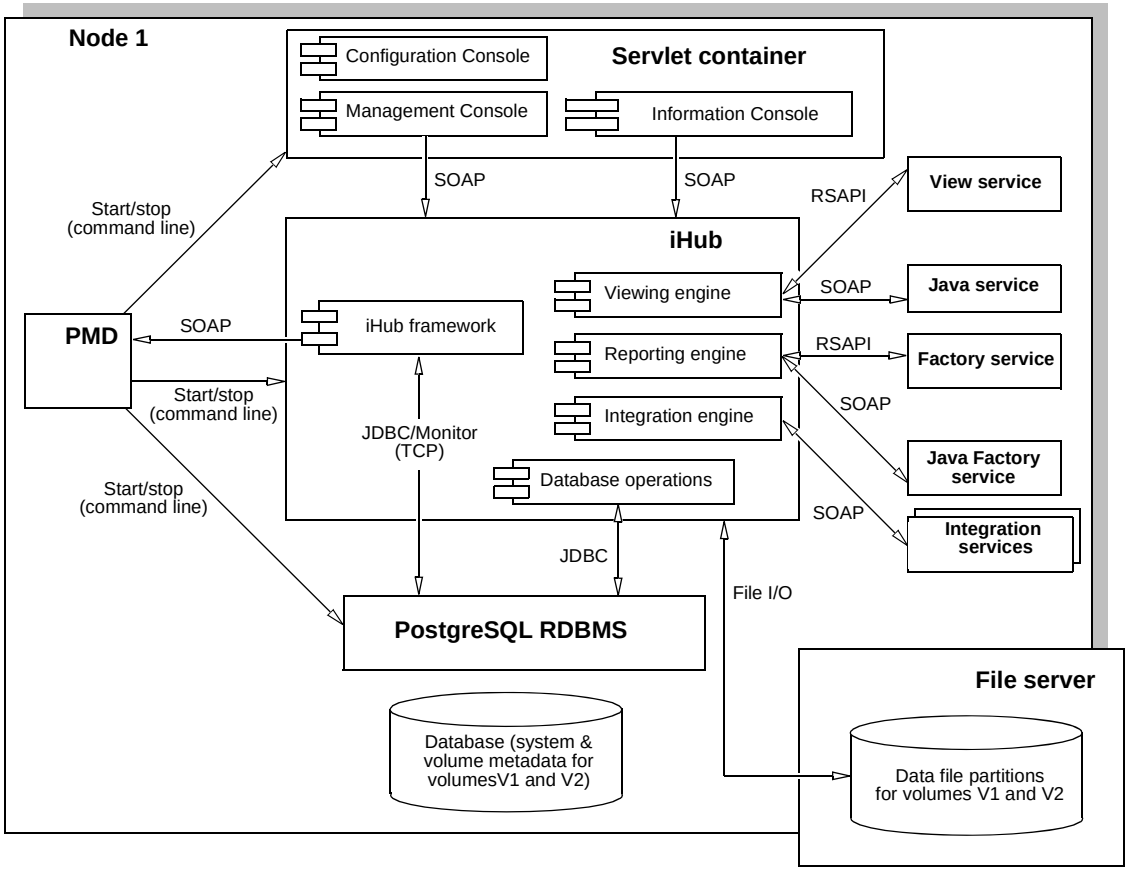


Figure 1-1 iHub process architecture for a stand-alone, two-volume, OOTB database

An IDAPI application uses a SOAP processor that serializes, or transforms, a remote procedure call by the application into an XML-based SOAP request to iHub to perform a web service. The application sends the message across the network using the HyperText Transfer Protocol (HTTP) transport layer.

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.

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 RDBMS 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.

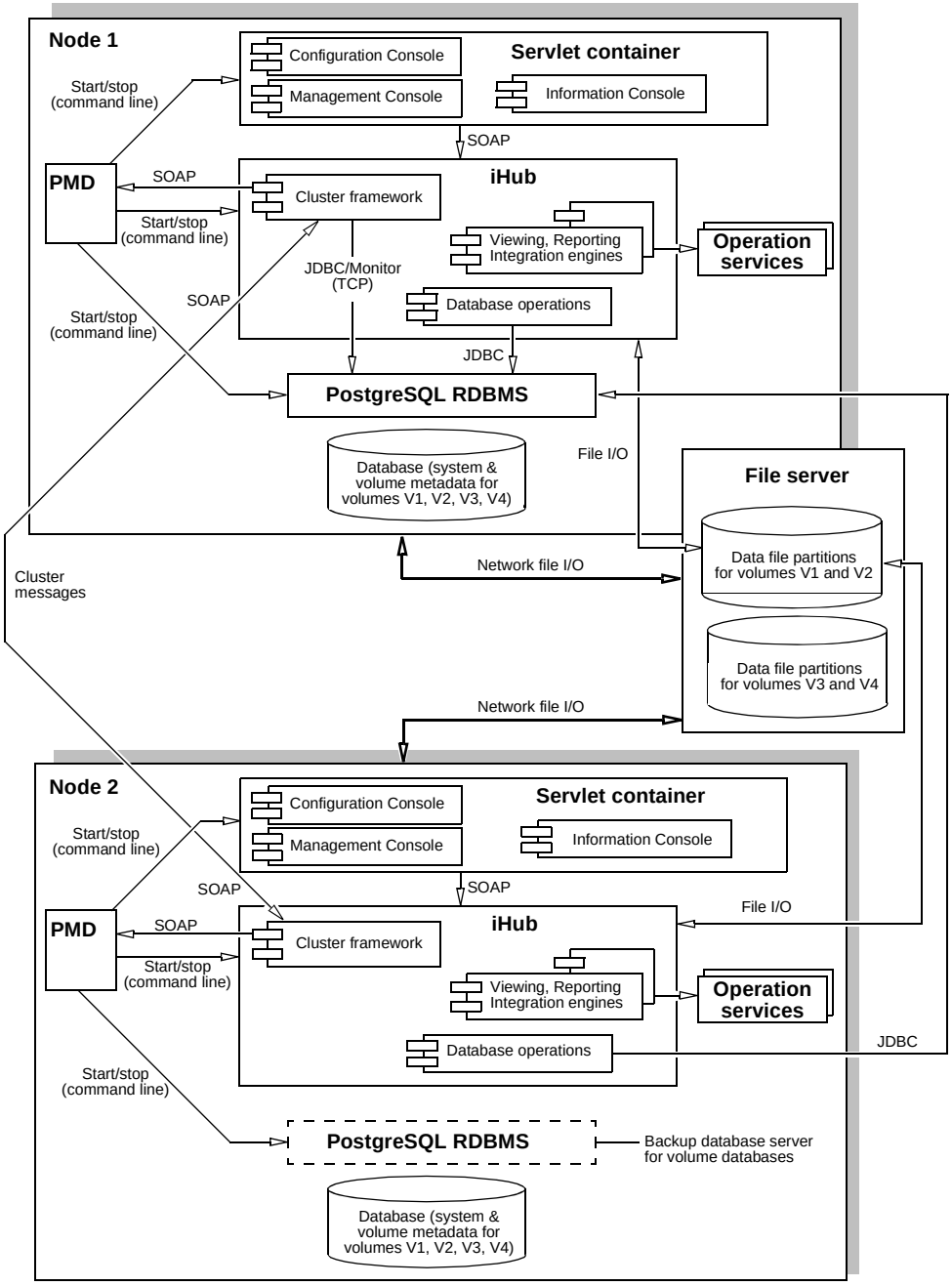


Figure 1-2 iHub process architecture for a clustered, two-node, four-volume, OOTB database

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 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 RDBMS, 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 a dedicated user account for installing and running iHub

Before installing iHub, create a dedicated Linux user account for installing, running, and administering iHub. Having a dedicated user account isolates iHub-specific issues and events on a machine, making it easier to administer the environment. Use the same level of security that your site exercises for other system administrator and root accounts.

Installation of iHub under the root account is not recommended and not supported when using the PostgreSQL RDBMS. The PostgreSQL RDBMS must run using an unprivileged user ID to prevent compromising system security. If installed under the root account, the default installation is unable to set up the required iHub metadata schemas and Encyclopedia sample volume.

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, refer to Chapter 10, "Backing up an Encyclopedia volume," later in this book.

When installing 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 iHub, you can download iHub from BIRT Exchange at the following location:

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

About X frame buffer

Xvfb is an X Windows server that has neither a graphics card nor a physical graphics display. BIRT iHub uses the X server for font-rendering information and to generate graphics in documents. Typically, an X server requires a graphics card and physical graphics display on the iHub machine, but you can use Xvfb in place of these components.

The Xvfb software installed with BIRT iHub includes Type 1 fonts. Actuate maps these fonts to Microsoft Windows fonts for consistent graphics rendering on the various platforms.

The Xvfb software uses:

- X libraries installed on the BIRT iHub machine

  If you choose to install Xvfb, the installation script searches for the required libraries and displays a message if the install script cannot find the required libraries on the machine.

- Variables set to the path of the Xvfb libraries
  - XVFBDISPLAY variable in start_srvr.sh.
  - display_value in pmd11.sh.

  To view and print the documents from BIRT iHub, you need to set these variables only if you install Xvfb software.

The DISPLAY environment variable specifies the X Windows server used by the iHub machine. For example, if the iHub machine is running X Windows, it sets DISPLAY to the local machine:

```
# setenv DISPLAY :0.0
```

If you use a separate machine as the X Windows server, specify the machine name in the environment variable DISPLAY. The following example sets DISPLAY to use an X Windows server on a machine named urup:

```
# setenv DISPLAY urup:0.0
```

The original source code for Xvfb is included as a component of X11R6, but not in earlier X Window system releases.

Installing X frame buffer

Actuate distributes Xvfb, and installation and configuration of Xvfb is a BIRT iHub installation option in this environment. To use Xvfb in HP-UX, you must install Xvfb before you install BIRT iHub.

About libstdc++

The libstdc++ library is a prerequisite for installing Actuate BIRT iHub on Linux systems. This library is present by default on most systems. If it is not present, the administrator must install it before installing iHub.

When installing BIRT iHub on a Linux machine, the following message may appear if you have the 64-bit version of the c/c++ run-time installed:

```
Error: An error occurred in the license reading program. Please
   make sure you have all the recommended patches and the right
   c/c++ runtime environment installed on this machine.
```

If this message appears, install the following 32-bit c++ run-time libraries:

```
compat-libstdc++-33.i686
libstdc++.i686
```

About run levels

The iHub installation process requires running Linux at run level 5. This level supports networking and multi-user mode with a graphical window manager. Run level 5 is typically the default on most Linux operating system distributions.

About Openmotif

On Linux platforms, the Openmotif bundle is required and must be installed before installing BIRT iHub. BIRT iHub needs libXm.so.3 or libXm.so.4, which are part of Openmotif 2.2 and 2.3, respectively. If the BIRT iHub installation is unable to locate the required libXm.so.x library, create a symbolic link, as shown in the following example:

```
ln -s libXm.so.3 libXm.so.4
```

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 BIRT iHub

The following section describes how to install a new, stand-alone instance of BIRT iHub in the Linux operating system.

How to perform a new installation of BIRT iHub in Linux

To reduce network traffic, you can install BIRT iHub on the same host machine as the iHub system database. You can also install BIRT iHub and the metadata database on a different machine to distribute processing across multiple machines.

Actuate recommends running the installation procedure from an account created exclusively for iHub administration. To install iHub, perform the following steps:

1. Download the required files from the FTP software distribution site. Extract the files.
2. To install the server files, execute the isinstall script:

   ```
   sh ./isinstall.sh
   ```

   The script displays a series of prompts. Respond to the prompts as described in the following procedures.
3. The license agreement appears, as shown in Figure 2-1.
4. Read the license agreement, then press Enter to continue the installation. At the prompt, type y for yes if you accept the licensing terms, as shown in Figure 2-2.
5. The introduction to the installation appears, as shown in Figure 2-3.

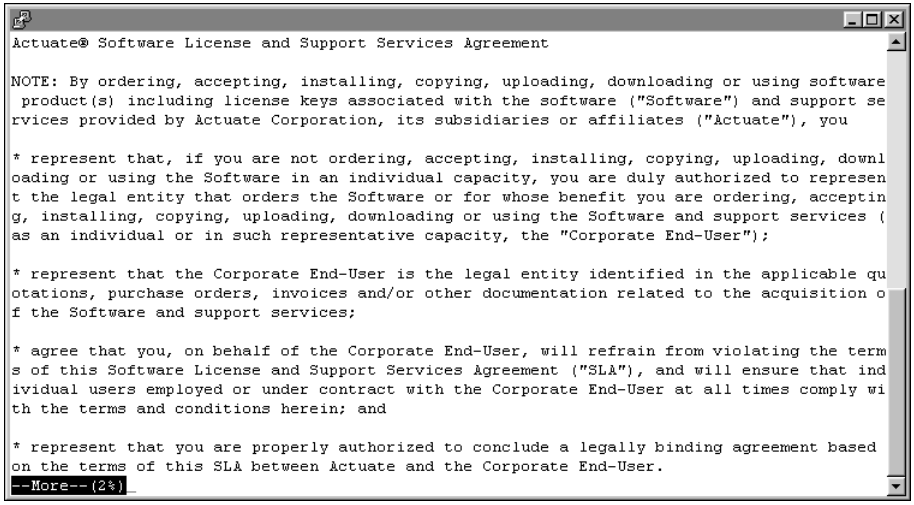


Figure 2-1 Reviewing the license agreement

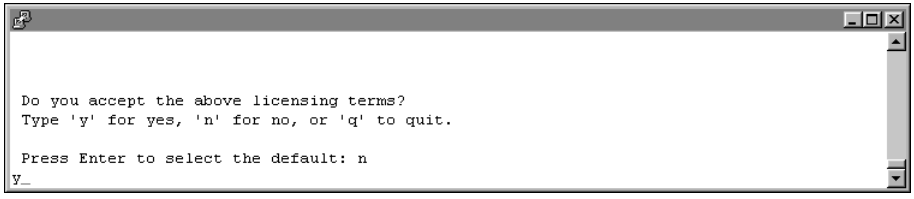


Figure 2-2 Accepting the licensing terms

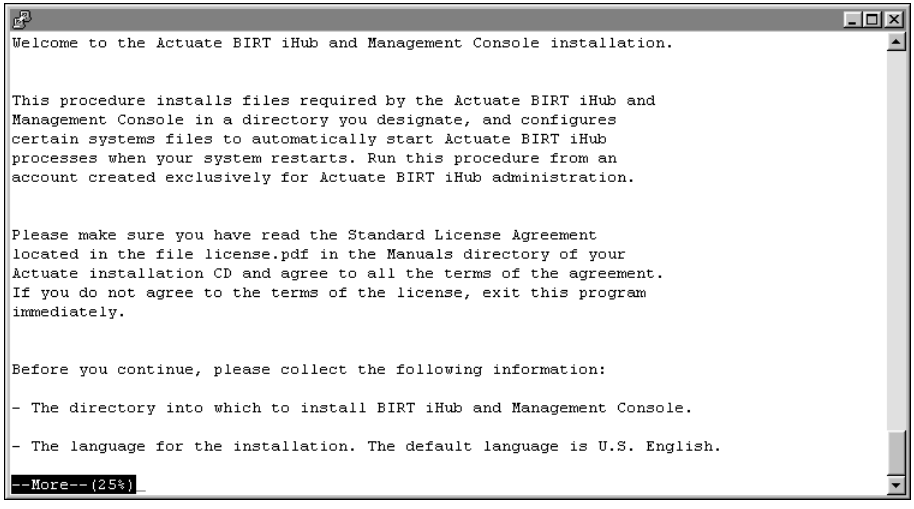


Figure 2-3 Reviewing the introductory information

6 Press Return or Enter after finishing the review of the introductory information, as shown in Figure 2-4.

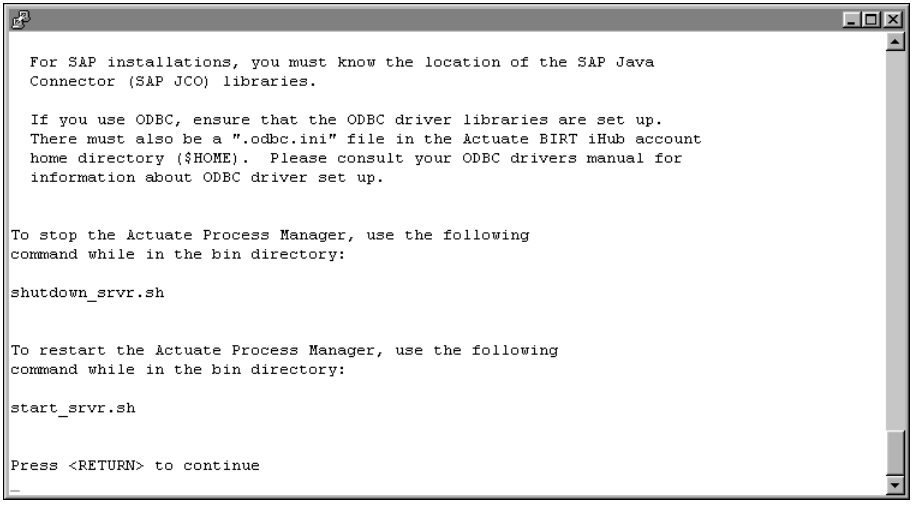


Figure 2-4 Finishing the review of introductory information

7 Press Enter to accept the default installation directory for BIRT iHub binaries, as shown in Figure 2-5. If installing BIRT iHub side-by-side with a Release 11 version of BIRT iServer to migrate from a Release 11 iServer to iHub, type a different directory, such as /home/actuate/iHub and press Enter. Alternatively, type a different directory and press Enter.

The installation program creates the Actuate directory in the chosen location. iHub uses this location to resolve the path to all binaries that it launches.

The default path is /home/actuate. If installing BIRT iHub side-by-side with a Release 11 version of BIRT iServer, the default path is /home/actuate /iHub. This documentation uses the environment variable AC_SERVER_HOME to refer to $HOME/AcServer in case the installer chooses a path that is different from the default path.

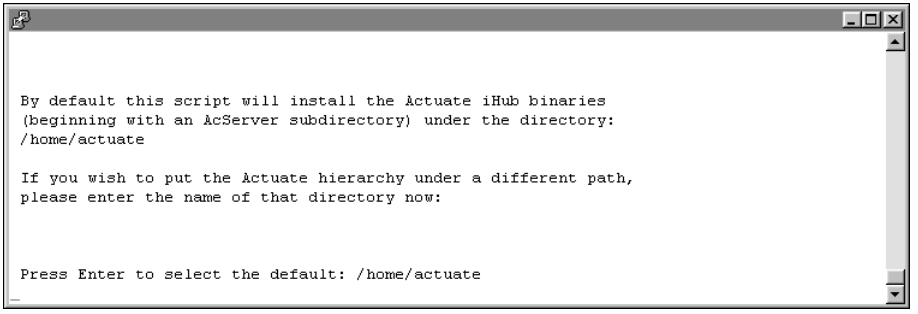


Figure 2-5 Specifying the installation directory

8 Press Enter to accept the default installation directory, AC_SERVER_HOME /data, for iHub data, as shown in Figure 2-6. Alternatively, choose a different directory for iHub data.

iHub uses this data location to store the iHub Encyclopedia volume data, including PostgreSQL metadata, logs, and other files.

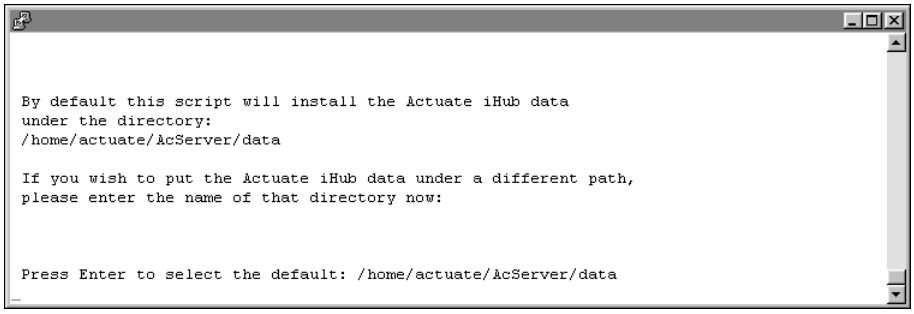


Figure 2-6 Specifying the data installation directory

9 Press Enter to accept the default option of creating the directory for data, as shown in Figure 2-7. Alternatively, type n for no, or q to quit, and press Enter.

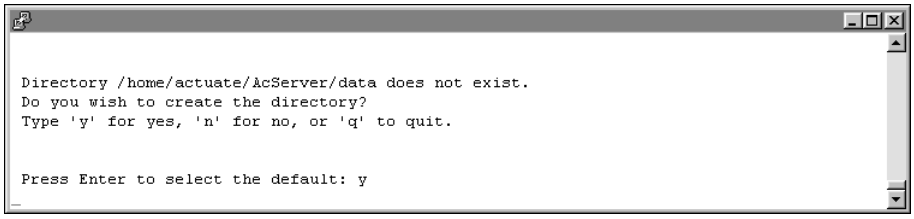


Figure 2-7 Creating the AC_DATA_HOME directory

10 The installer copies prerequisite files to the destination folder, as shown in Figure 2-8. After the prerequisite files are copied, the installation continues.

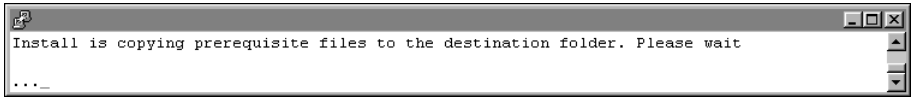


Figure 2-8 Copying prerequisite files

11 Press Enter to accept the default iHub component combination, which includes Management Console, as shown in Figure 2-9. Alternatively, choose a different component combination and press Enter.

12 Press Enter to accept the default stand-alone Server installation, as shown in Figure 2-10. Alternatively, choose a different type of iHub to install. For information on how to install an iHub cluster, see Chapter 5, "Installing a BIRT iHub cluster."

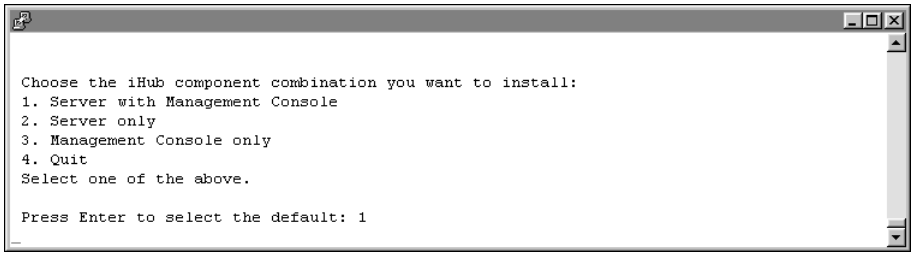


Figure 2-9 Choosing the components to install

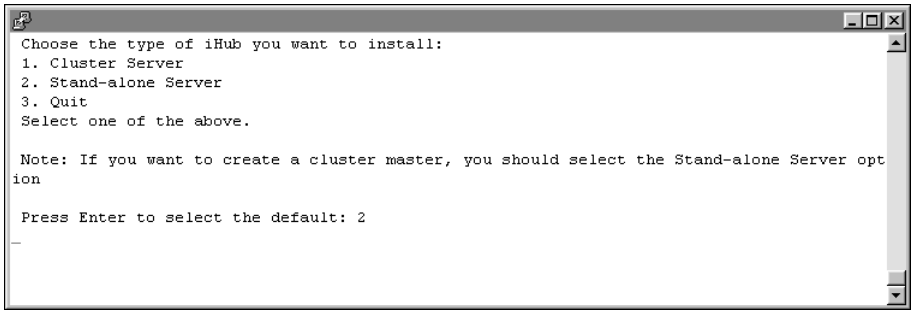


Figure 2-10 Specifying the type of iHub to install

13 Type a name to use for the BIRT iHub System name, as shown in Figure 2-11. 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.

Figure 2-11 Specifying the BIRT iHub System name

14 Press Enter to choose the default embedded PostgreSQL database to store the Encyclopedia volume metadata, as shown in Figure 2-12.

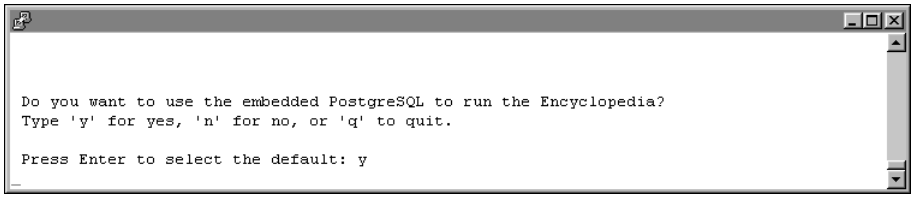


Figure 2-12 Choosing the embedded PostgreSQL

15 Press Enter to choose the default PostgreSQL superuser name, postgres, as shown in Figure 2-13. Alternatively, type a different PostgreSQL superuser name. This superuser administers the PostgreSQL relational database management system (RDBMS).

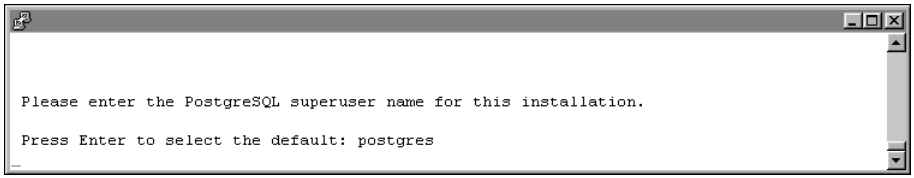


Figure 2-13 Choosing the PostgreSQL superuser name

16 Type a PostgreSQL superuser password that conforms to the password security policy requirements within your organization, then press Enter, as shown in Figure 2-14.

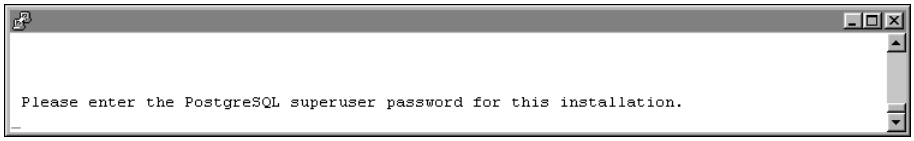


Figure 2-14 Typing the PostgreSQL superuser password

17 Re-enter the password for PostgreSQL superuser, then press Enter, as shown in Figure 2-15.

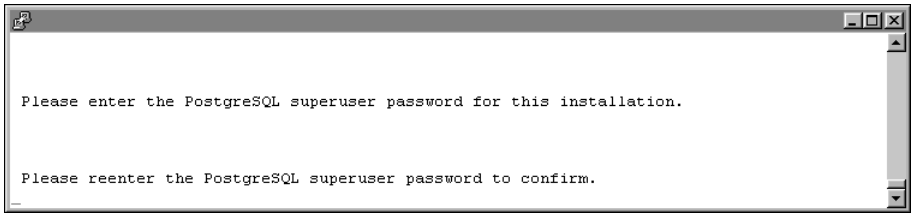


Figure 2-15 Re-entering the PostgreSQL superuser password

18 Press Enter to accept the default port on which PostgreSQL listens for requests, as shown in Figure 2-16.

If the install program detects that the default port number is in use, the install program displays the next highest port number available For example, if the default port is 8432, but another process is using that port, the install program displays the next available port, such as 8433.

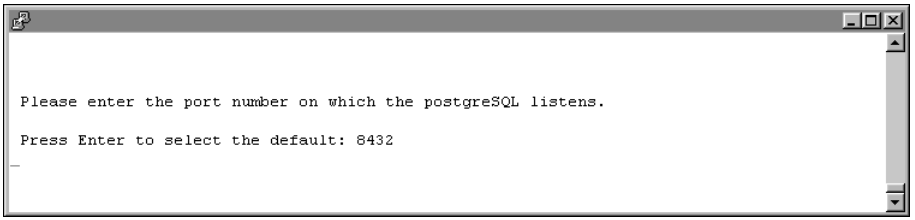


Figure 2-16 Choosing the PostgreSQL port number

19 Press Enter to select the default locale, which is English, as shown in Figure 2-17. Alternatively, select a different locale. If you do not see the locale for your region, type m for more and press Enter.

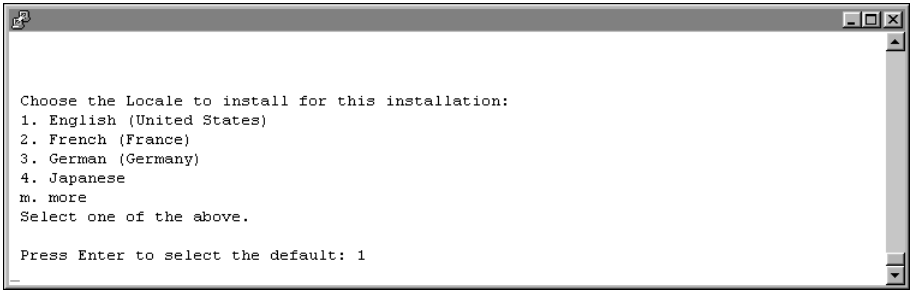


Figure 2-17 Specifying a locale

20 Press Enter to select the default time zone, which is America/Los_Angeles, as shown in Figure 2-18. Alternatively, select another time zone from the numbered list.

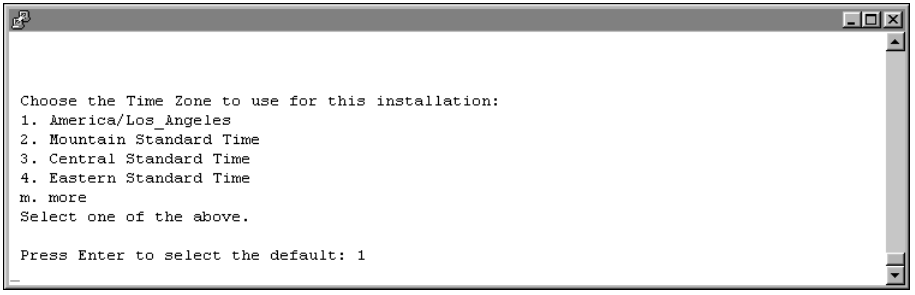


Figure 2-18 Specifying a time zone

21 To evaluate the product using the included evaluation software license press Enter, as shown in Figure 2-19. Alternatively, type 2, then type the path to a purchased license file.

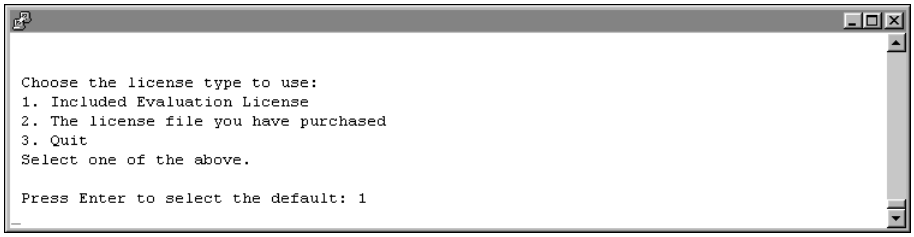


Figure 2-19 Specifying license type

22 Press Enter to accept the hostname of the machine that Management Console uses to contact the Process Management Daemon (PMD), as shown in Figure 2-20. Alternatively, type a different IP address.

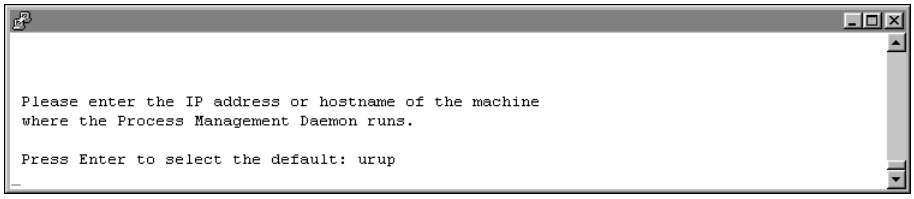


Figure 2-20 Specifying the hostname that Management Console uses to contact the PMD

23 Press Enter to accept the default port number, 8100, where Process Management Daemon (PMD) listens for requests, as shown in Figure 2-21. Alternatively, type a different port number and press Enter.

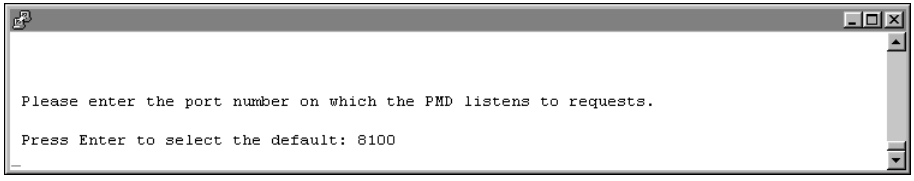


Figure 2-21 Specifying the port number on which the PMD listens

24 Press Enter to accept the default host name, the name of the machine on which iHub runs, as shown in Figure 2-22. Alternatively, type a different host name or IP address, then press Enter.

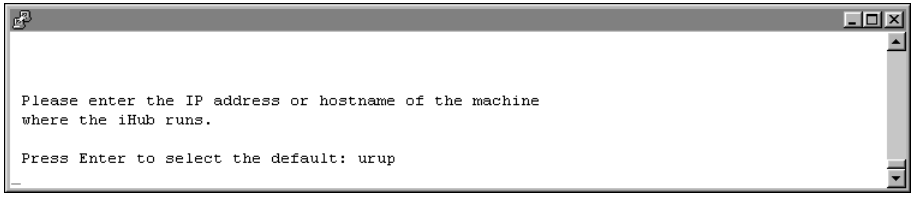


Figure 2-22 Specifying the machine on which the iHub runs

25 Press Enter to accept the default port number where iHub listens to requests, as shown in Figure 2-23. Alternatively, type a different port number and press Enter.

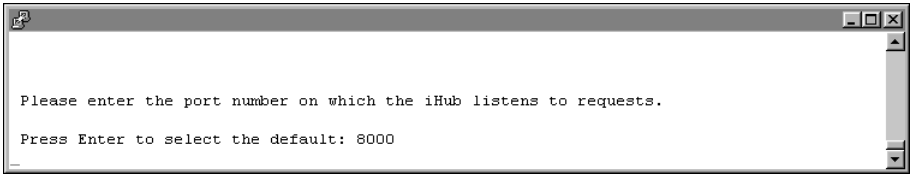


Figure 2-23 Specifying the port number on which the iHub listens

26 Specify the iHub administrator password, as shown in Figure 2-24. You use this password to log into the iHub Configuration Console.

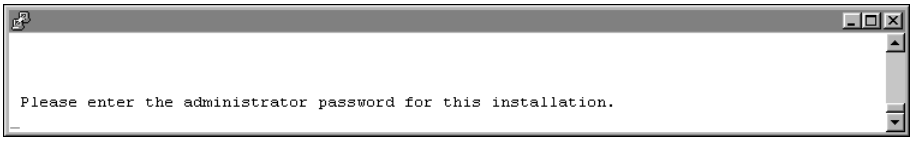


Figure 2-24 Specifying the iHub administrator password

27 Re-enter the password of the iHub administrator, as shown in Figure 2-25. You use this password to log in to Configuration Console.

Figure 2-25 Re-entering the iHub administrator password

28 Press Enter to accept the default option to use a volume name for the Encyclopedia, as shown in Figure 2-26. Alternatively, type n for no to not use a volume name for the Encyclopedia, or q to quit the installation.

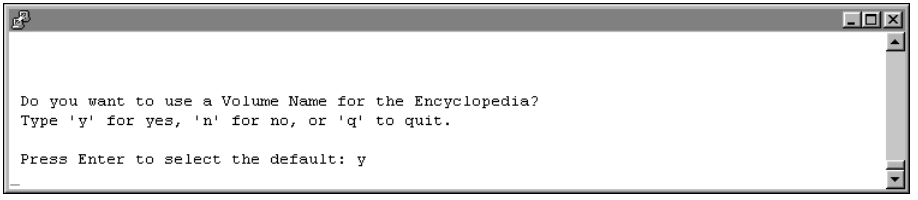


Figure 2-26 Specifying whether to use an Encyclopedia volume name

29 Press Enter to accept the default Encyclopedia volume name, the BIRT iHub System name, as shown in Figure 2-27. Alternatively, type a different Encyclopedia volume name.

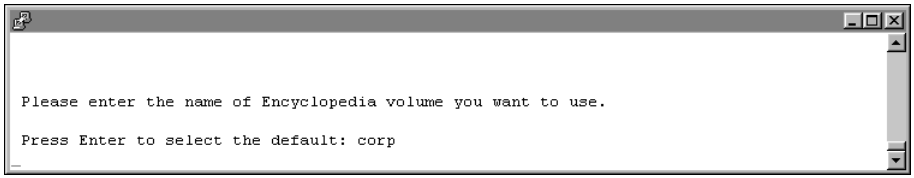


Figure 2-27 Specifying the Encyclopedia volume name

30 Press Enter to accept the default option to start iHub automatically, as shown in Figure 2-28. Alternatively, type n for no.

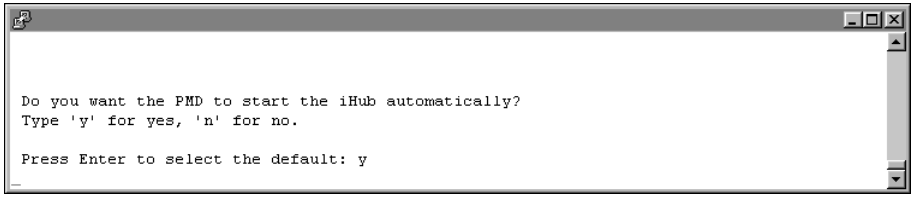


Figure 2-28 Specifying whether to start iHub automatically

31 Press Enter to accept the default option to not integrate LDAP with iHub, as shown in Figure 2-29. Alternatively, you can edit the setting.

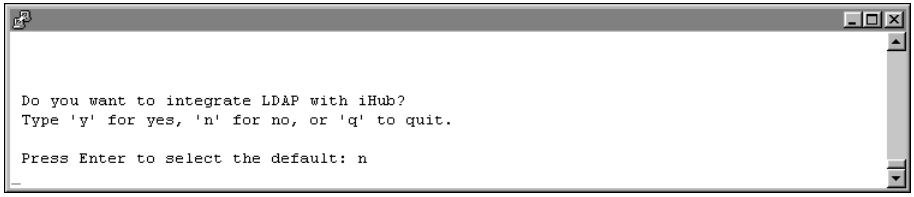


Figure 2-29 Specifying whether to integrate LDAP with iHub

32 Press Enter to accept the default option to not use any database drivers/clients, as shown in Figure 2-30. Alternatively, type y for yes, and specify the database drivers/clients you want to use.

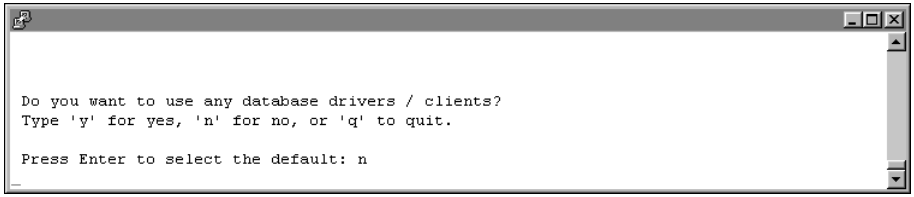


Figure 2-30 Specifying whether to use database drivers/clients

33 Specify what kind of X-Server you want to use, if any. To accept the default, press Enter, as shown in Figure 2-31.

34 Press Enter to accept the hostname of the machine that Management Console uses to contact the Process Management Daemon (PMD), as shown in Figure 2-32. Alternatively, type a different IP address.

35 Press Enter to accept the default port number, 8100, on which the Process Management Daemon (PMD) listens for requests from Management Console, as shown in Figure 2-33. Alternatively, type a different port number.

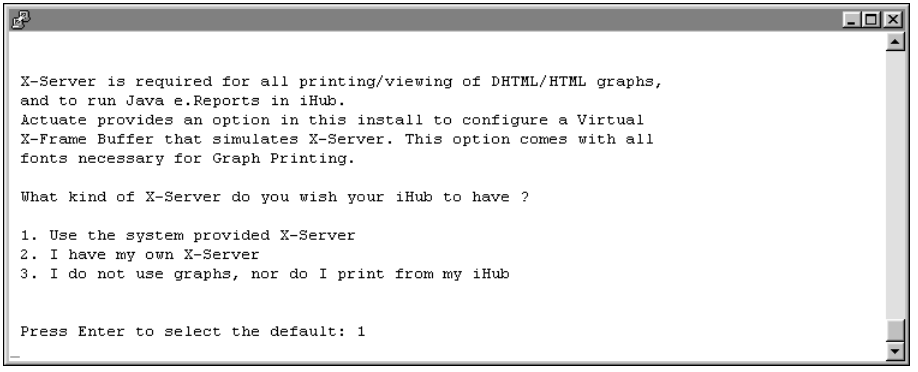


Figure 2-31 Specifying what kind of X-Server to use, if any

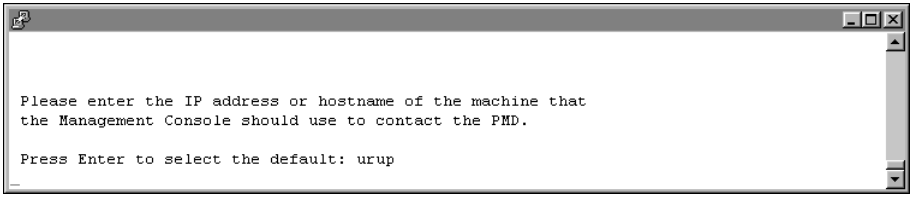


Figure 2-32 Specifying the hostname that Management Console uses to contact the PMD

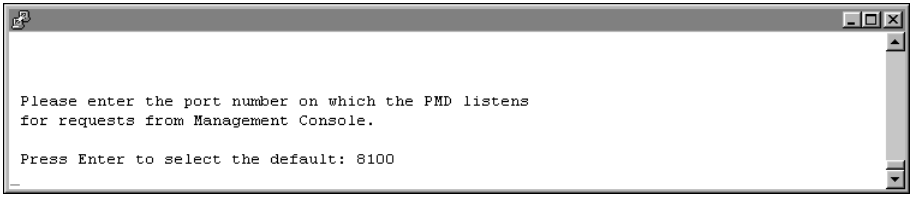


Figure 2-33 Specifying the port number for the PMD to listen for requests from Management Console

36 Press Enter to accept the hostname or enter the IP address of the machine that Management Console uses to contact iHub, as shown in Figure 2-34. Alternatively, type a different IP address.

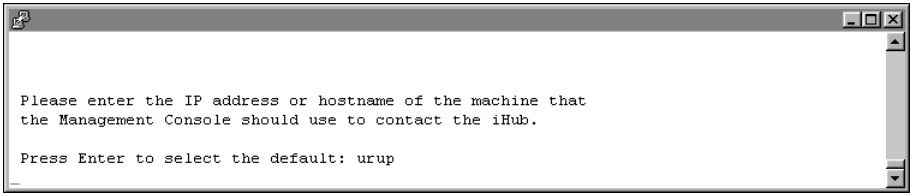


Figure 2-34 Specifying the hostname that Management Console uses to contact iHub

37 Press Enter to accept the default port number, 8000, on which iHub listens for requests from Management Console, as shown in Figure 2-35. Alternatively, type a different port number.

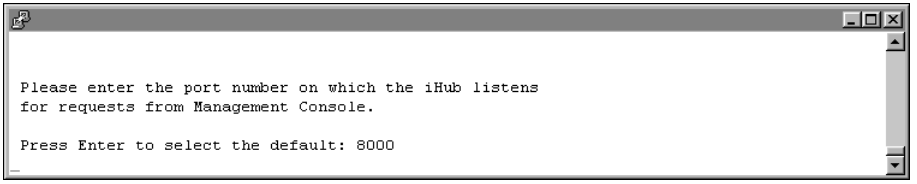


Figure 2-35 Specifying the port number on which iHub listens for requests from Management Console

38 Press Enter to accept the name of the default Encyclopedia volume to use with Management Console, as shown in Figure 2-36. Alternatively, type a different name for the Encyclopedia volume.

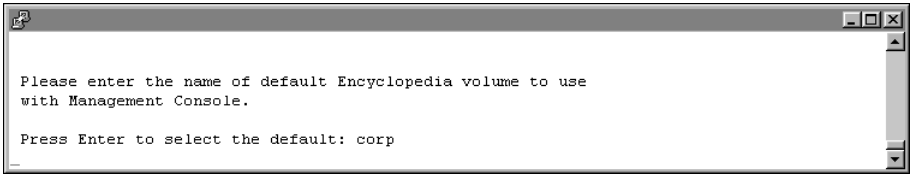


Figure 2-36 Specifying the name of the default Encyclopedia volume

39 Press Enter to accept the default name, acadmin, for the HTTP server context root for Management Console configuration, as shown in Figure 2-37. Alternatively, type a different name.

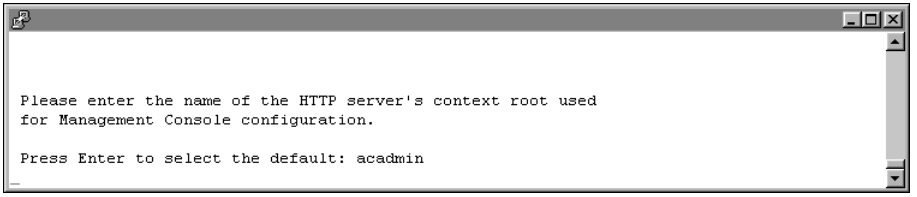


Figure 2-37 Specifying the name of the HTTP server context root

40 Press Enter to accept the default HTTP port number, 8900, on which the application container listens to requests, as shown in Figure 2-38. Alternatively, choose a different port.

You connect to the port from your browser when accessing various features of iHub.

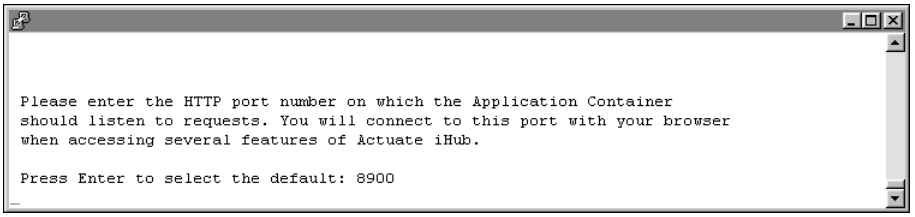


Figure 2-38 Specifying the application container listening port number

41 Review the settings, as shown in Figure 2-39, then specify whether you accept the settings. Press Enter to accept the default, y for yes. Alternatively, type n for no, or q to quit.

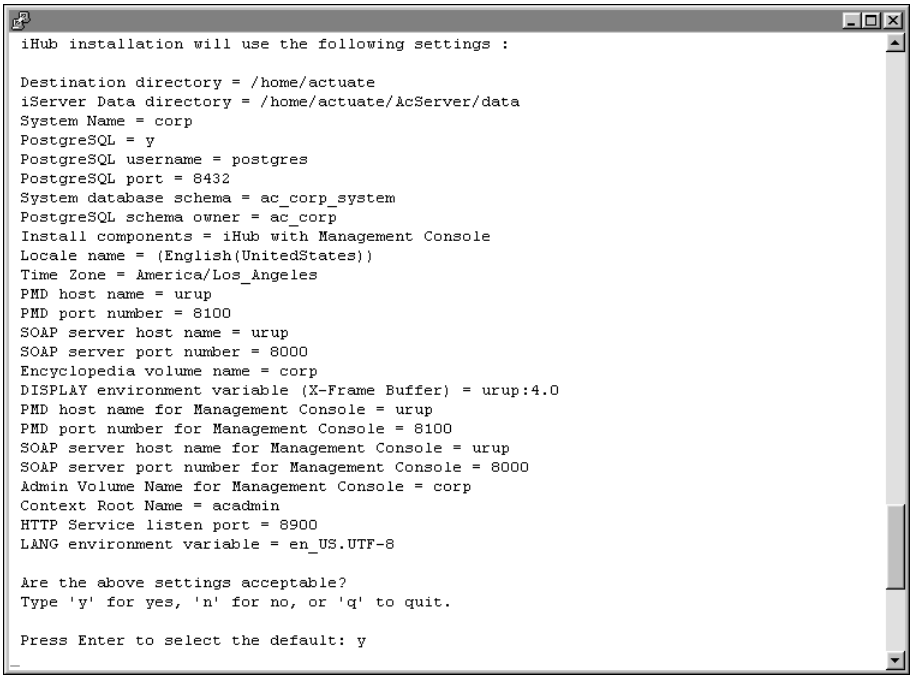


Figure 2-39 Reviewing the installation settings

42 The installation program installs iHub, displaying an indicator that shows the progress of the installation, as shown in Figure 2-40.

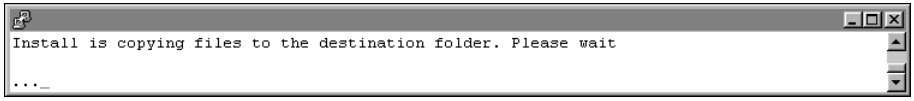


Figure 2-40 Viewing iHub installation progress

43 At the end of the installation, the program asks if you want to start iHub. Accept the default, y for yes, to start iHub, as shown in Figure 2-41.

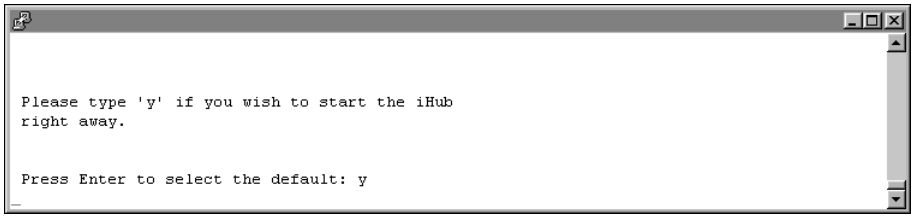


Figure 2-41 Specifying whether to start iHub

44 When the installation program finishes, it provides additional information about localization, logging in using an account with root permissions to start iHub, and installing online help and manuals, as shown in Figure 2-42.

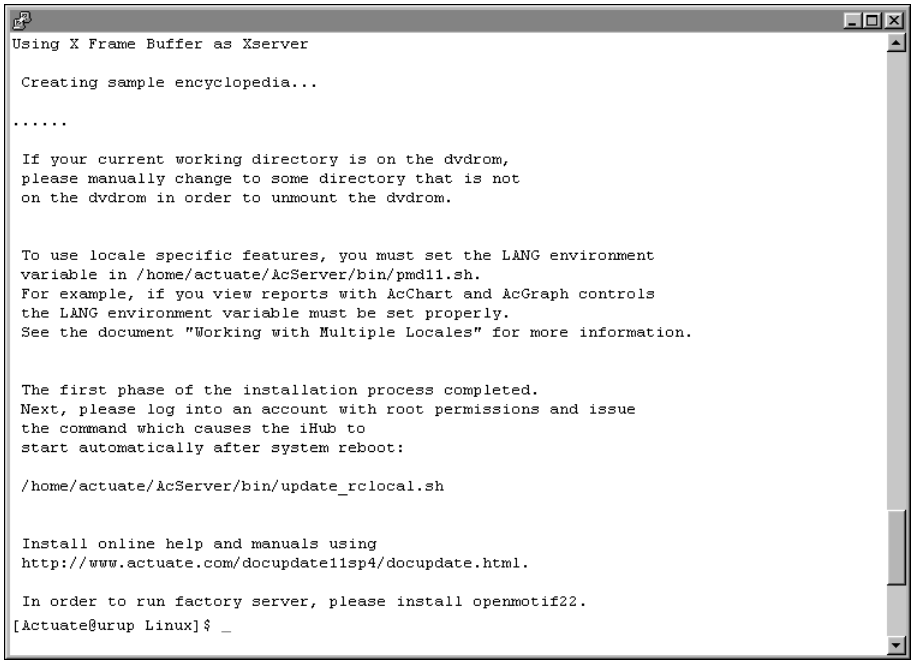


Figure 2-42 Viewing information about localization, logging in, and installing online help

Accessing Information, Management, and Configuration Consoles

After the installation program finishes running, open a browser to log in to the following BIRT iHub consoles to perform user and administrator tasks:

- Information Console

  Perform tasks such as accessing folders and viewing designs and documents.

  To access Information Console, open a browser manually and enter the following URL, as shown in Figure 2-43:

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

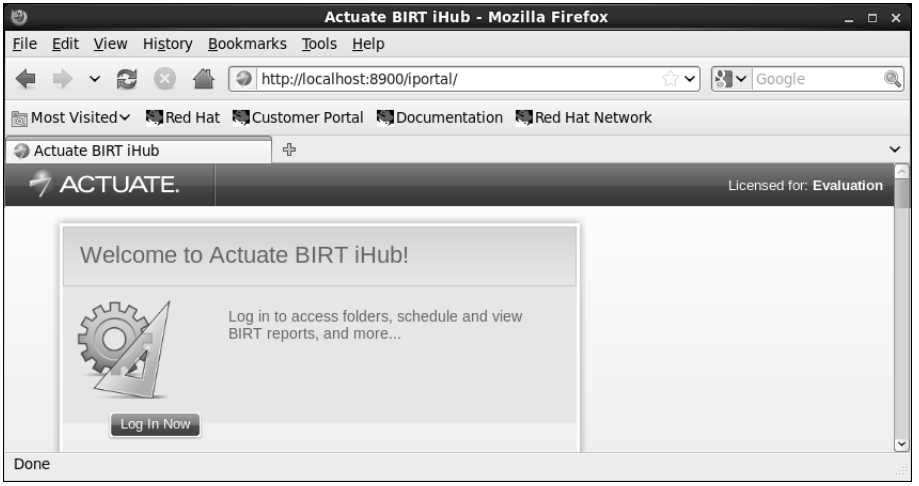


Figure 2-43 Viewing Welcome to Actuate BIRT iHub

- Management Console

  Set up user accounts and schedule or run a design.

  To access Management Console, open a browser manually and enter the following URL, as shown in Figure 2-44:

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

Figure 2-44 Logging in to Management Console

- Configuration Console
  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 2-45:

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

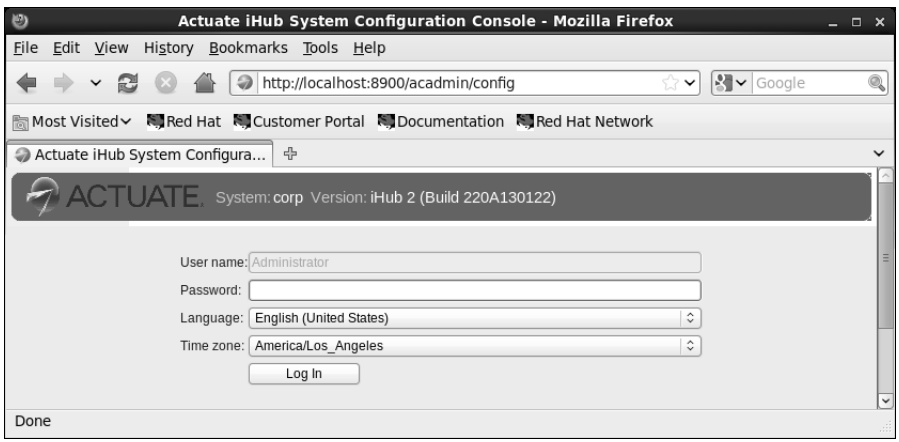


Figure 2-45 Logging in to Configuration Console

When starting PostgreSQL for BIRT 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 BIRT iHub

To stop 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:

```
sh ./shutdown_srvr.sh
```

To start 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:

```
sh ./startsrvr.sh
```

How to stop and start PostgreSQL for BIRT iHub

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:

```
sh ./stoppostgresql.sh
```

To restart PostgreSQL for BIRT 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:

```
sh ./startpostgresql.sh
```

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 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:

```
AC_SERVER_HOME/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 Actuate report 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 Linux, specify the CLASSPATH in the Process Management Daemon (PMD) startup script, pmd11.sh.

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 report 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.

  Encyclopedia volume data is located separately 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 a dedicated user account for installing and running BIRT iHub

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.

If you exercise the same control over the user account for BIRT iHub as your site exercises for other system administrator and root accounts, you can maintain the same level of security for BIRT iHub. Actuate does not recommend installing iHub under the root account since the PostgreSQL server must be started and maintained under an unprivileged user ID to prevent compromising system security. If installed under the root account, the default installation is unable to set up the PostgreSQL schemas and Actuate Encyclopedia sample volume.

Creating the 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 a 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 encoding 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 Oracle as an alternative data store.

Installing an Encyclopedia volume that uses an alternative database

The following procedures use a pre-existing Oracle database and schema as the example. During the iHub installation, the administrator provides the following installation details and any related credentials:

- External Oracle database host name or IP address, such as urup
- iHub Encyclopedia volume name, which by default is the machine name, in this case, urup
- Oracle database iHub system schema name, such as ac_corp_system
- Oracle database Encyclopedia volume schema name, such as ac_corp
- Oracle database iHub application user name, such as iserver

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

To install iHub, perform the following steps:

1. Download the required files from the FTP software distribution site. Extract the files.
2. To install the server files, execute the following isinstall script:

   ```
   sh ./isinstall.sh
   ```

   The script displays a number of prompts. Respond to the prompts as described in the following procedure.

3 The license agreement appears, as shown in Figure 3-1.

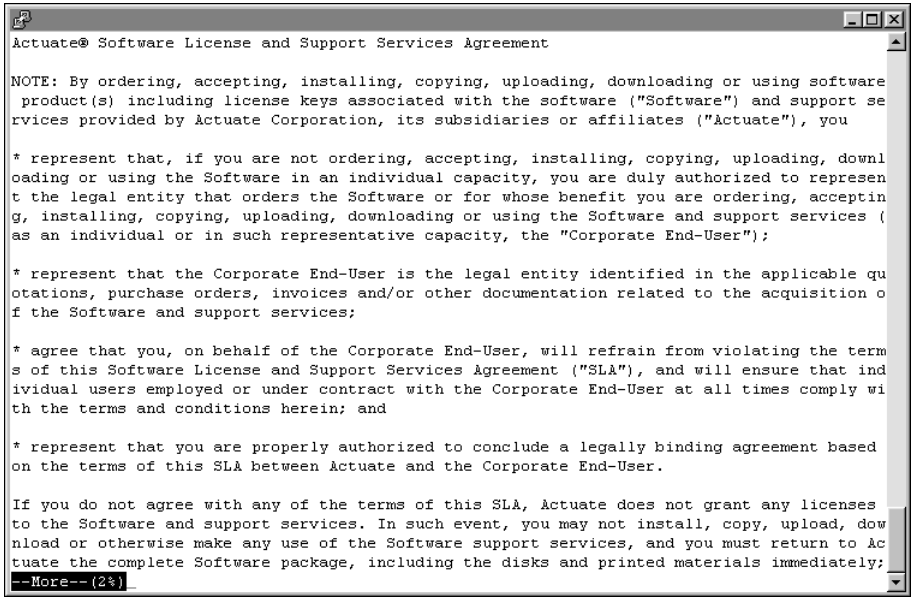


Figure 3-1 Reviewing the license agreement

4 Read the license agreement and press Enter to continue the installation. At the prompt, type y for yes if you accept the licensing terms, as shown in Figure 3-2.

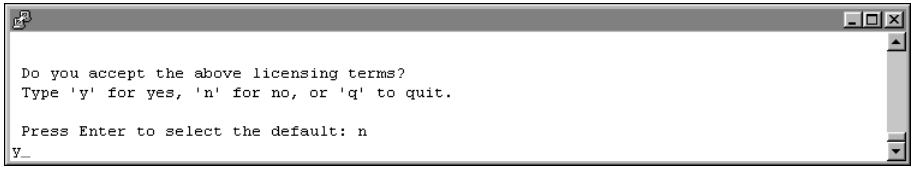


Figure 3-2 Accepting the licensing terms

5 The introduction to the installation appears, as shown in Figure 3-3.

6 Press Enter after reviewing the introductory information, as shown in Figure 3-4.

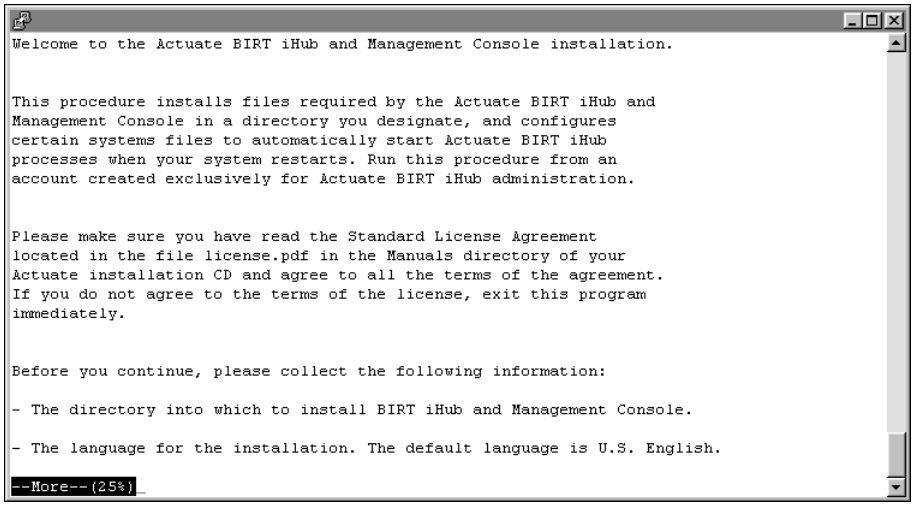


Figure 3-3 Reviewing the introductory information

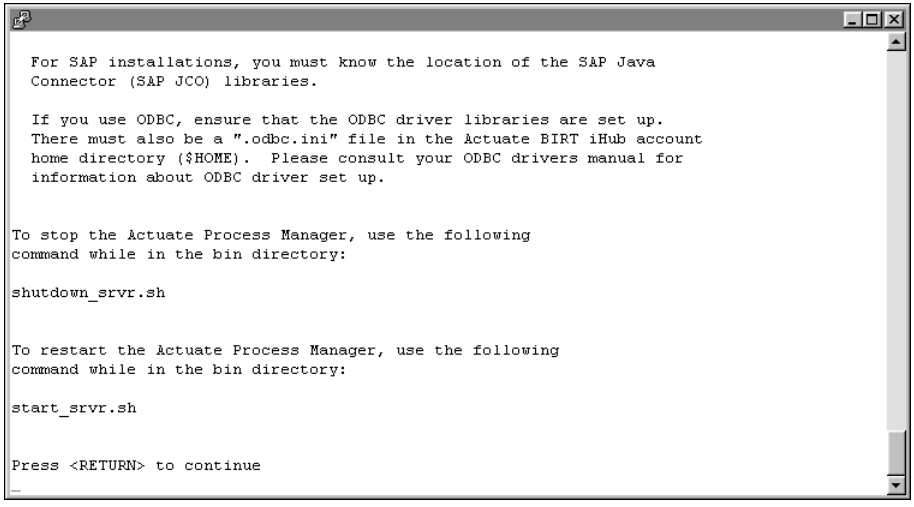


Figure 3-4 Finishing the review of introductory information

7 Press Enter to accept the default location for the installation, as shown in Figure 3-5. Alternatively, type a different directory and press Enter.

The installation program creates the AcServer directory in the chosen location and installs the files.

iHub uses this location to resolve paths to all the binaries that it launches. The default path for this location is $HOME/AcServer, which is referred to in the iHub documentation by the environment variable, AC_SERVER_HOME.

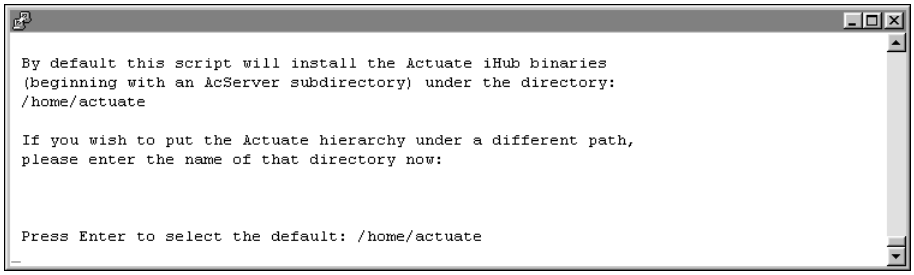


Figure 3-5 Specifying the installation directory

8 Press Enter to accept the default installation directory, AC_SERVER_HOME /data, for iHub data, as shown in Figure 3-6. Alternatively, choose a different directory for iHub data.

iHub uses this data location to store the iHub Encyclopedia volume data, logs, and other files. The default path is AC_SERVER_HOME/data, which is referred to in the iHub documentation by the environment variable AC_DATA_HOME.

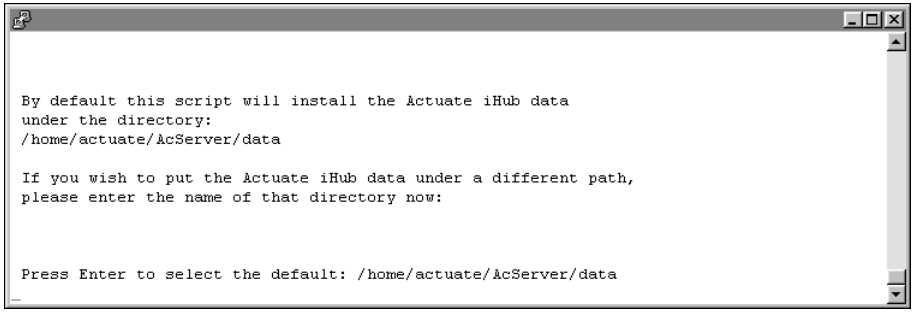


Figure 3-6 Specifying the data installation directory

9 Press Enter to accept the default option of creating the directory for data, as shown in Figure 3-7. Alternatively, type n for no, or q to quit, and press Enter.

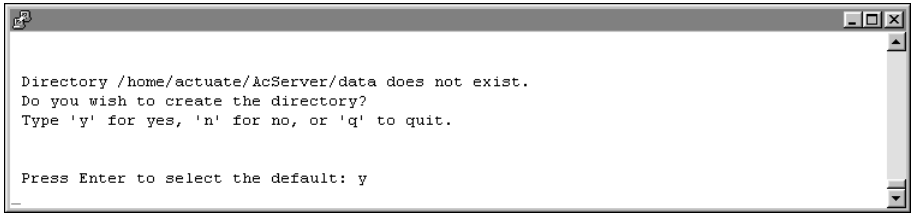


Figure 3-7 Creating the AC_DATA_HOME directory

10 The installer copies prerequisite files to the destination folder, as shown in Figure 3-8. After copying the prerequisite files, the installation continues.

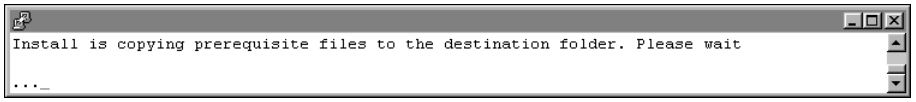


Figure 3-8 Copying prerequisite files

11 Press Enter to accept the default iHub component combination, iHub with Management Console, as shown in Figure 3-9. Alternatively, choose a different iHub component combination and press Enter.

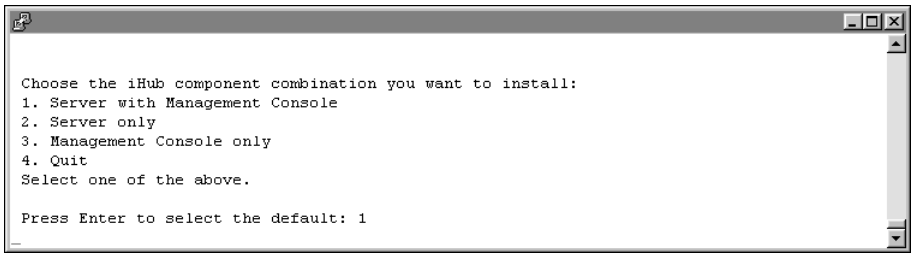


Figure 3-9 Choosing the iHub component combination

12 Press Enter to accept the default stand-alone iHub installation, as shown in Figure 3-10. Alternatively, choose a different type of iHub to install.

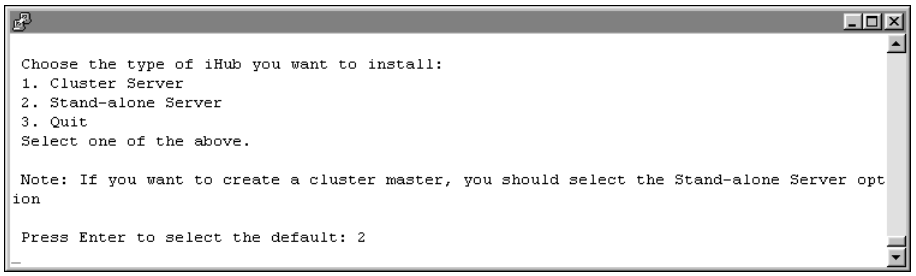


Figure 3-10 Specifying the type of iHub to install

13 Type a name to use for the BIRT iHub System name, as shown in Figure 3-11. 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.

Figure 3-11 Specifying the BIRT iHub System name

14 Type n for no, and press Enter, as shown in Figure 3-12. You do not want to install the embedded PostgreSQL database if you are using an alternative database such as Oracle to store Encyclopedia volume metadata.

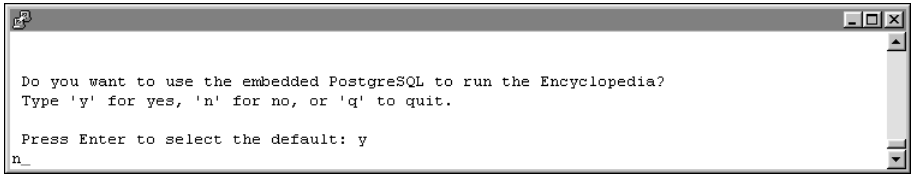


Figure 3-12 Choosing to not use Embedded PostgreSQL

15 Type 2 to choose Oracle as the external database to work with iHub Encyclopedia and press Enter, as shown in Figure 3-13.

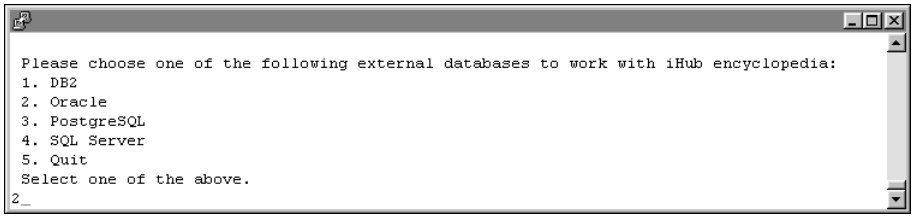


Figure 3-13 Choosing an external database for iHub Encyclopedia

16 Specify the external Oracle database TNS server, if any, and press Enter, as shown in Figure 3-14. If there is no external Oracle database TNS Server, leave the field blank and press Enter.

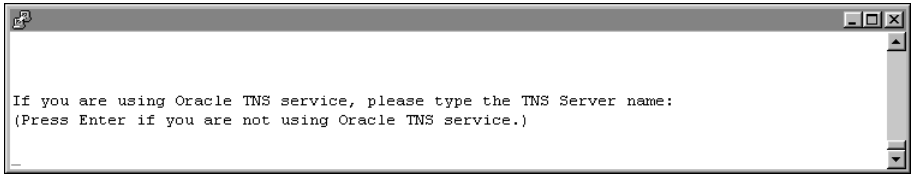


Figure 3-14 Specifying the external Oracle database TNS Server

17 Specify the name of the external Oracle database host name or IP address and press Enter, as shown in Figure 3-15.

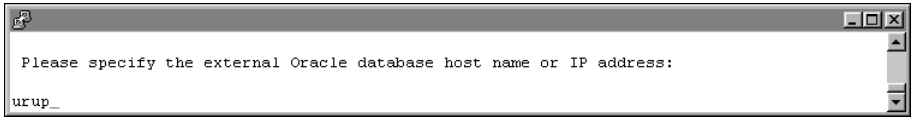


Figure 3-15 Specifying the external Oracle database server name

18 Specify the external Oracle database port number and press Enter, as shown in Figure 3-16. Typically, Oracle uses port 1521 as the database port.

Figure 3-16 Specifying the external Oracle database port

19 Specify the external Oracle database service name, such as orcl.actuate.com, that identifies the Oracle database server on which you want to install the Encyclopedia volume metadata and press Enter, as shown in Figure 3-17.

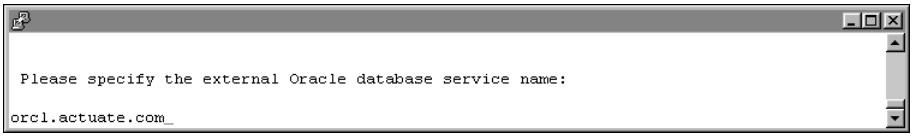


Figure 3-17 Specifying the external Oracle database service name

20 Specify the external Oracle database user name, such as iserver, and press Enter, as shown in Figure 3-18.

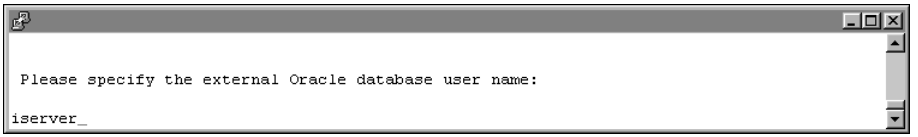


Figure 3-18 Specifying the external Oracle database user name

21 Specify the external Oracle database user password and press Enter, as shown in Figure 3-19.

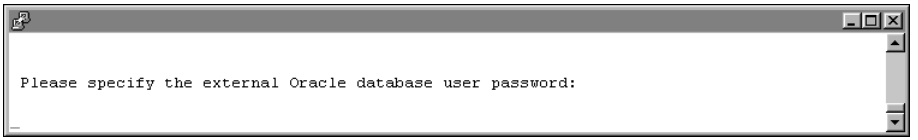


Figure 3-19 Specifying the external Oracle database user password

22 Re-enter the external Oracle database user password and press Enter, as shown in Figure 3-20.

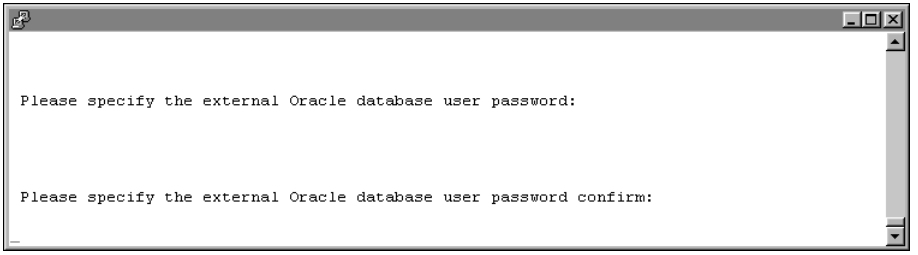


Figure 3-20 Confirming the external Oracle database user password

23 Specify the system database schema for iHub, such as ac_corp_system, and press Enter, as shown in Figure 3-21.

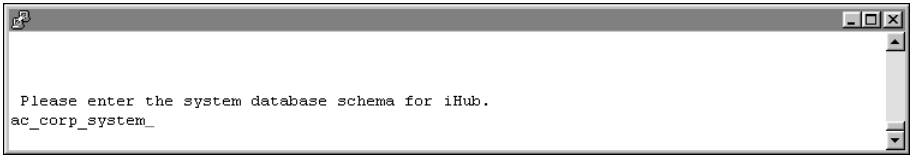


Figure 3-21 Specifying the system database schema

24 Specify the System database schema password, and press Enter, as shown in Figure 3-22.

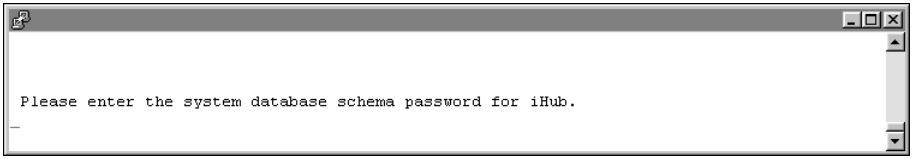


Figure 3-22 Specifying the system database schema password

25 Re-enter the system database schema password and press Enter, as shown in Figure 3-23.

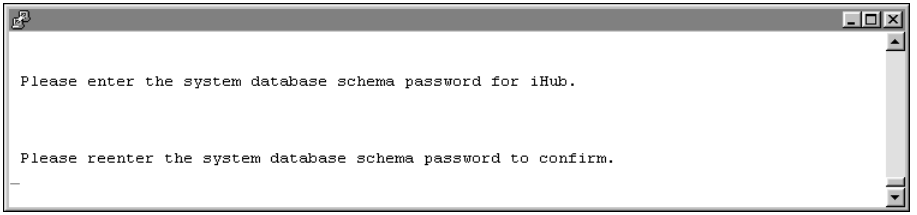


Figure 3-23 Confirming the system database schema password

26 Specify the external Oracle database schema for the Encyclopedia volume, such as ac_corp, and press Enter, as shown in Figure 3-24.

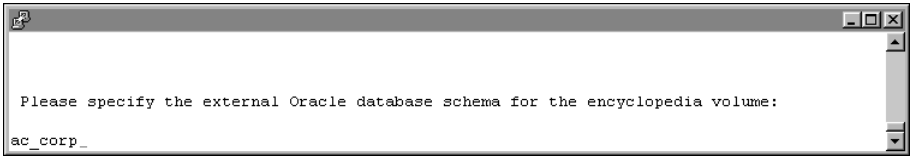


Figure 3-24 Specifying the schema for the Encyclopedia volume

27 Specify the external Oracle database schema password, and press Enter, as shown in Figure 3-25.

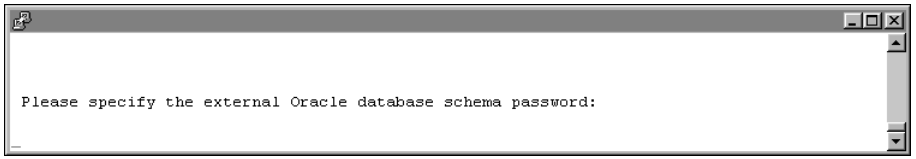


Figure 3-25 Specifying the external Oracle database schema password

28 Re-enter the external Oracle database schema password and press Enter, as shown in Figure 3-26.

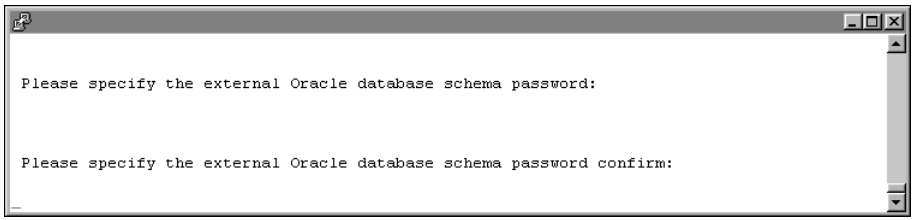


Figure 3-26 Confirm the external Oracle database schema password

29 Press Enter to select the default locale, which is English (United States), as shown in Figure 3-27. Alternatively, select a different locale. If you do not see the locale for your region, type m for more and press enter.

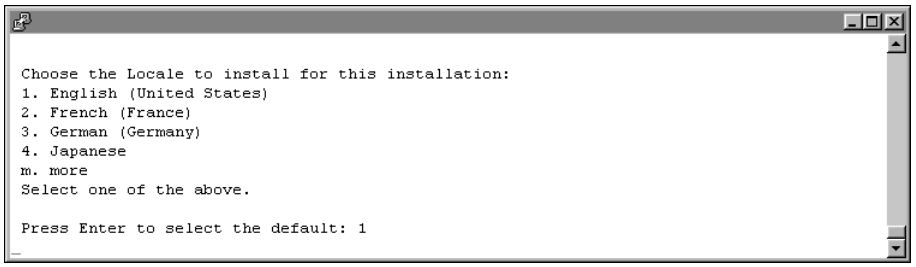


Figure 3-27 Specifying a locale

30 Press Enter to select the default time zone, America/Los_Angeles, as shown in Figure 3-28. Alternatively, select another time zone from the numbered list.

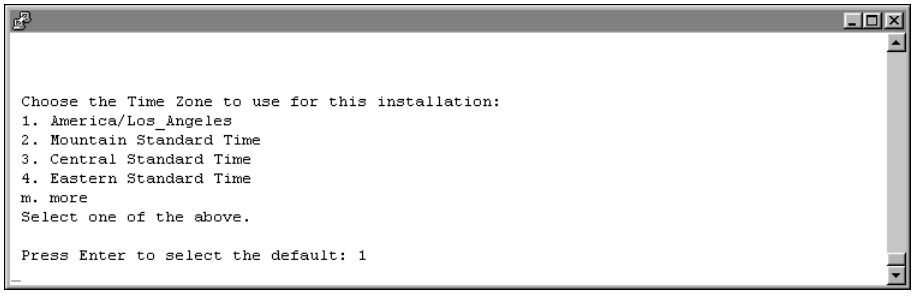


Figure 3-28 Specifying a time zone

31 To evaluate the product using the included evaluation software license, press Enter, as shown in Figure 3-29. Alternatively, type 2, then type the path to the license file you purchased.

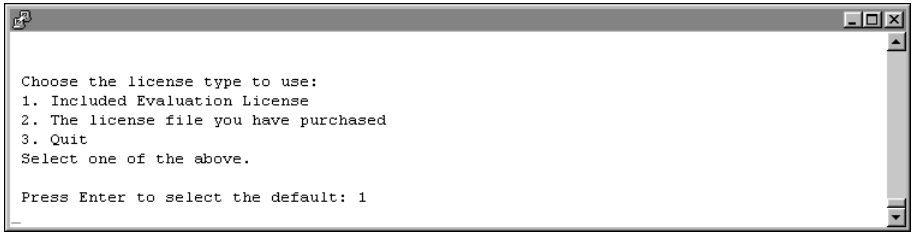


Figure 3-29 Specifying license type

32 Press Enter to accept the hostname of the machine that Management Console uses to contact the Process Management Daemon (PMD), as shown in Figure 3-30. Alternatively, type a different IP address.

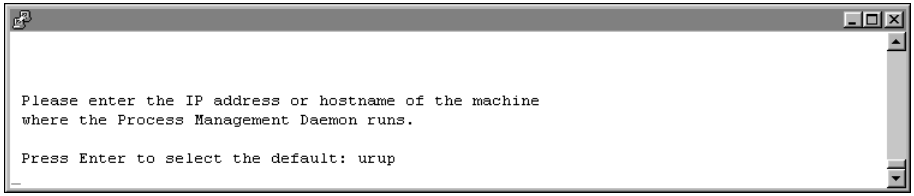


Figure 3-30 Specifying the hostname that Management Console uses to contact the PMD

33 Press Enter to accept the default port number, 8100, where Process Management Daemon (PMD) listens for requests, as shown in Figure 3-31. Alternatively, type a different port number.

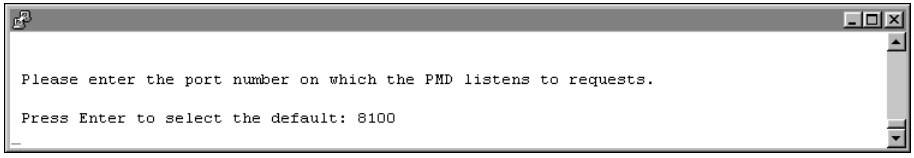


Figure 3-31 Specifying the port number on which the PMD listens

34 Press Enter to accept the default host name, the name of your machine, as shown in Figure 3-32. Alternatively, type a different IP address.

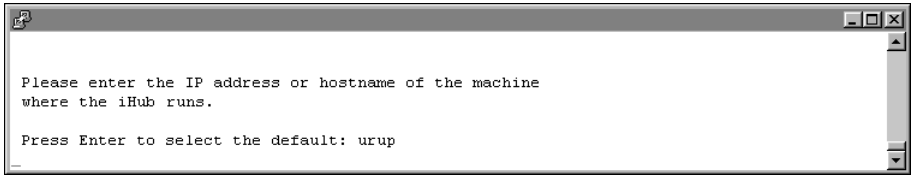


Figure 3-32 Specifying the machine on which the iHub runs

35 Press Enter to accept the default port number, 8000, where iHub listens for requests, as shown in Figure 3-33. Alternatively, type a different port number.

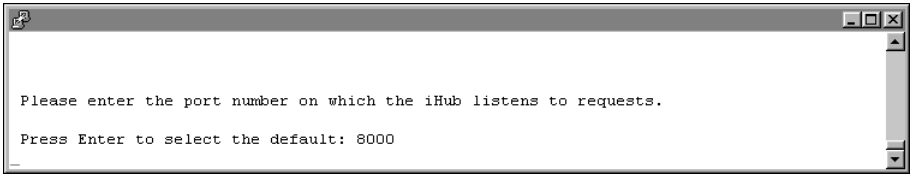


Figure 3-33 Specifying the port number on which the iHub listens

36 Specify the iHub administrator password, as shown in Figure 3-34.

You use this password to log in to the iHub Configuration Console.

Figure 3-34 Specifying the iHub administrator password

37 Re-enter the password of the iHub administrator, as shown in Figure 3-35.

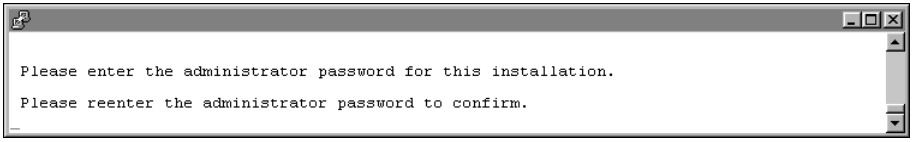


Figure 3-35 Re-entering the iHub administrator password

38 Press Enter to accept the default option to use a volume name for the Encyclopedia, as shown in Figure 3-36. Alternatively, type n for no to not use a volume name for the Encyclopedia, or q to quit the installation.

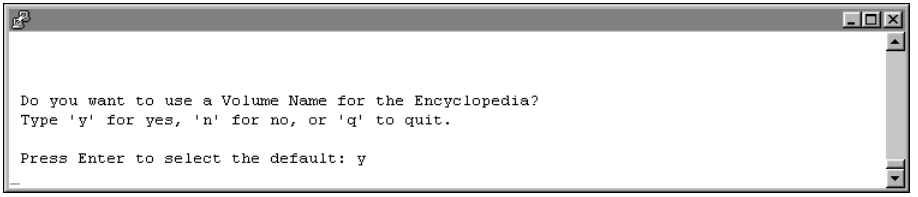


Figure 3-36 Specifying whether to use a volume name

39 Press Enter to accept the default Encyclopedia volume name, the name of your machine, as shown in Figure 3-37. Alternatively, type a different Encyclopedia volume name.

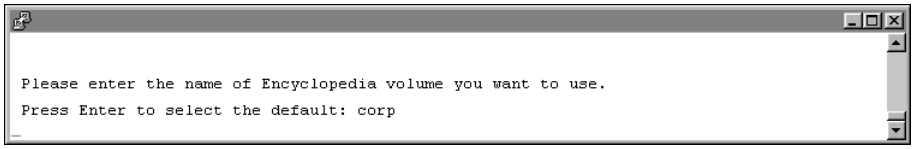


Figure 3-37 Specifying the Encyclopedia volume name

40 Press Enter to accept the default option to start iHub automatically, as shown in Figure 3-38. Alternatively, type n for no.

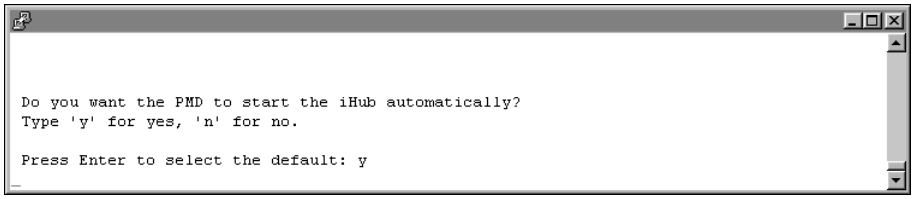


Figure 3-38 Specifying whether to start iHub automatically

41 Press Enter to accept the default option of not integrating LDAP with iHub, as shown in Figure 3-39. Alternatively, type n for no, or q to quit the installation.

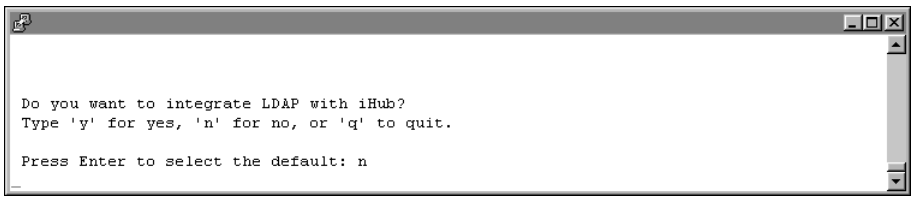


Figure 3-39 Specifying whether to integrate LDAP with iHub

42 Press Enter to accept the default option to not use any database drivers/clients, as shown in Figure 3-40. Alternatively, type y for yes, and specify the database drivers/clients you want to use.

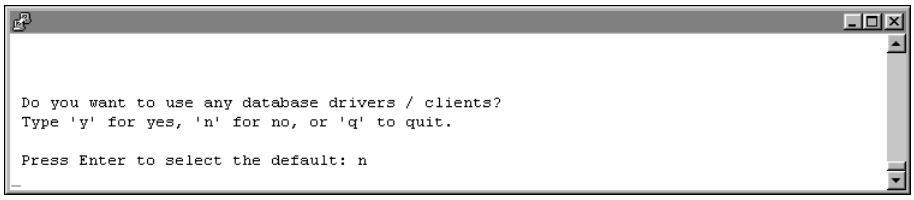


Figure 3-40 Specifying whether to use database drivers/clients

43 Specify what kind of X-Server you want to use, if any. To accept the default, press Enter, as shown in Figure 3-41.

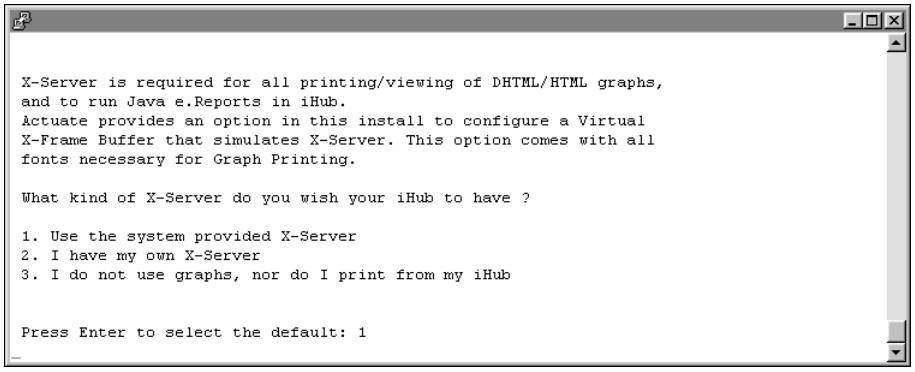


Figure 3-41 Specifying what kind of X-Server to use, if any

44 Press Enter to accept the default hostname of the machine that Management Console uses to contact the Process Management Daemon (PMD), as shown in Figure 3-42. Alternatively, type a different IP address.

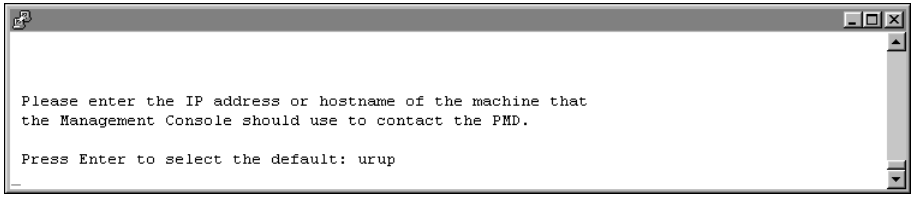


Figure 3-42 Specifying the machine host name that Management Console uses to contact the PMD

45 Press Enter to accept the default port number, 8100, on which the Process Management Daemon (PMD) listens for requests from Management Console, as shown in Figure 3-43. Alternatively, type a different port number.

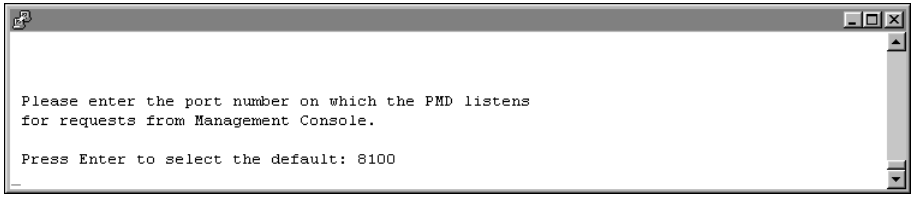


Figure 3-43 Specifying the port number for the PMD to listen for requests from Management Console

46 Press Enter to accept the default hostname, the name of your machine, that Management Console uses to contact iHub, as shown in Figure 3-44. Alternatively, type a different IP address.

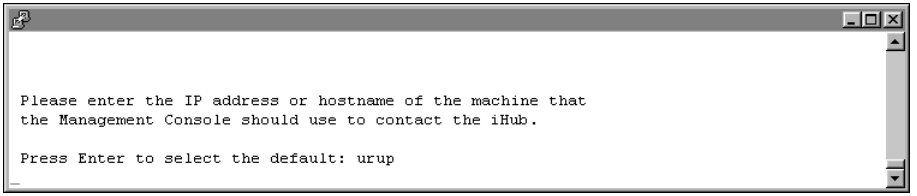


Figure 3-44 Specifying the name Management Console uses to contact iHub

47 Press Enter to accept the default port number, 8000, on which iHub will listen for requests from Management Console, as shown in Figure 3-45. Alternatively, type a different port number.

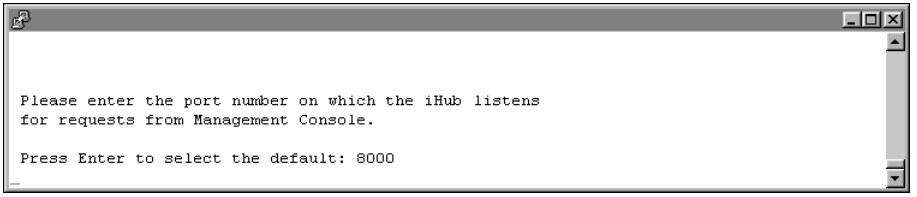


Figure 3-45 Specifying the port number for listening for requests from Management Console

48 Press Enter to accept the default name of the Encyclopedia volume to use with Management Console, as shown in Figure 3-46. Alternatively, type a different name for the Encyclopedia volume.

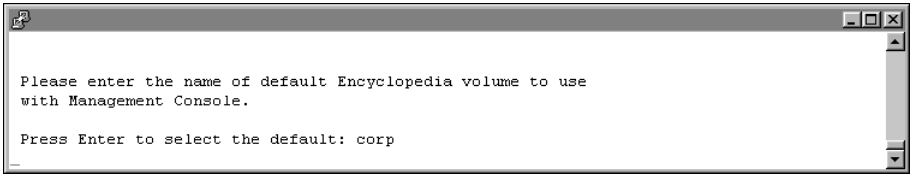


Figure 3-46 Specifying the name of the default Encyclopedia volume

49 Press Enter to accept the default name, acadmin, for the HTTP server's context root, as shown in Figure 3-47. Alternatively, type a different name.

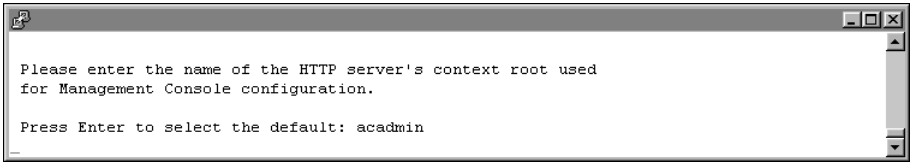


Figure 3-47 Specifying the name of the HTTP server context root

50 Press Enter to accept the default port number, 8900, on which the application container listens for requests, as shown in Figure 3-48. Alternatively, choose a different port.

You connect to the port from your browser when accessing various iHub features.

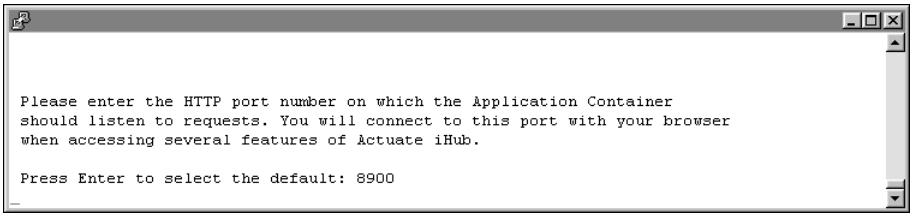


Figure 3-48 Specifying the application container listening port number

51 Review the settings, as shown in Figure 3-49, then specify whether to accept the settings. Press Enter to accept the default, y for yes. Alternatively type n for no, or q to quit.

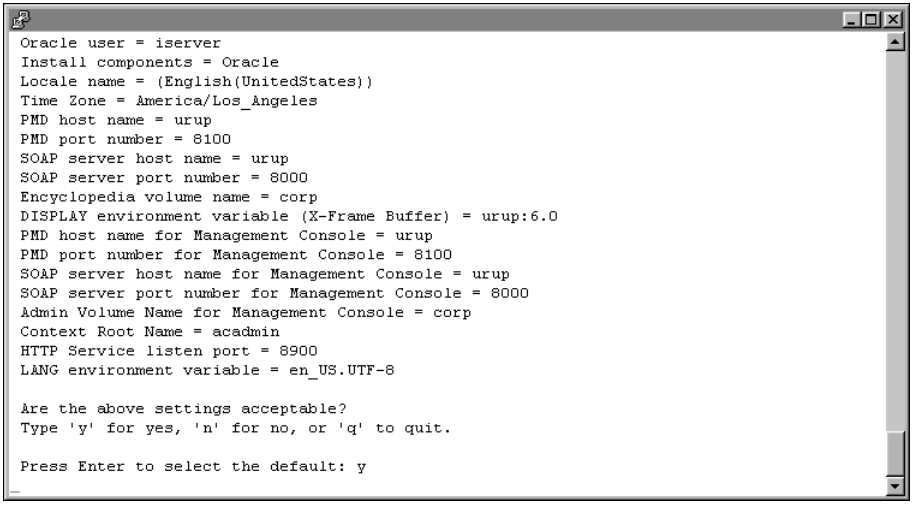


Figure 3-49 Reviewing your settings

52 The installation program installs iHub, displaying an indicator that shows the progress of the installation, as shown in Figure 3-50.

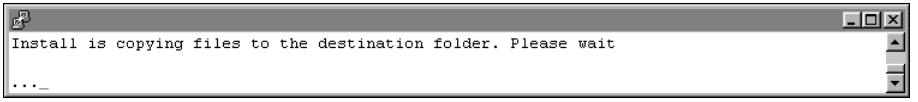


Figure 3-50 Copying iHub files to your destination folder

53 At the end of the installation, the program asks if you want to start iHub. Accept the default, y for yes, to start the Process Management Daemon (PMD), as shown in Figure 3-51.

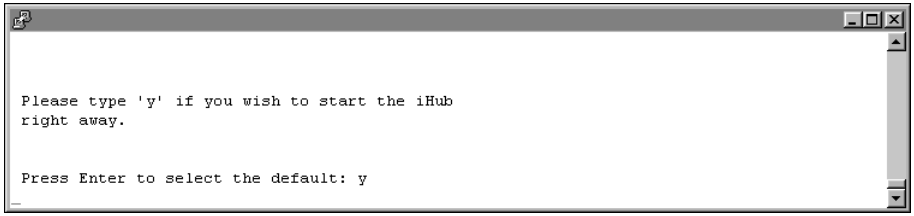


Figure 3-51 Specifying whether to start iHub

54 The installation program provides additional information about localization, logging in using an account with root permissions to start iHub, and installing online help and manuals, as shown in Figure 3-52.

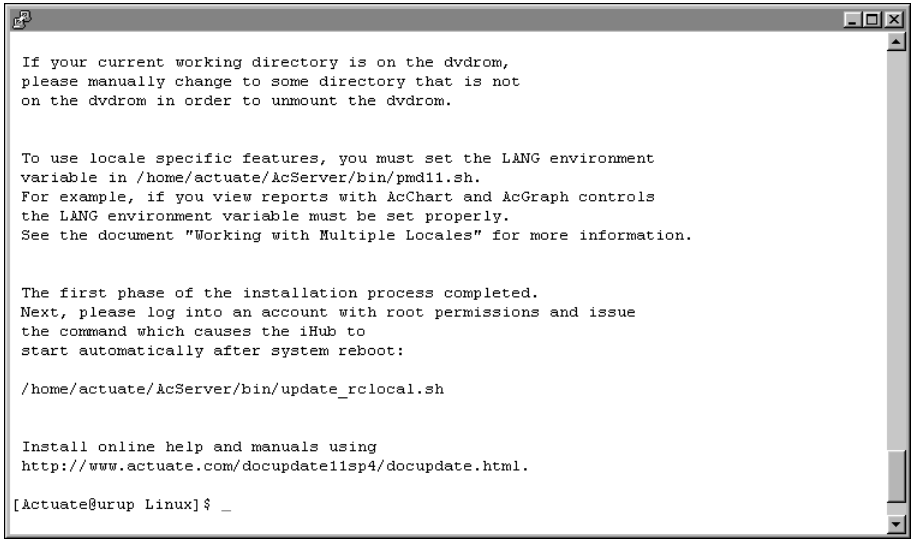


Figure 3-52 Specifying information about localization, logging in, and installing online help

After the installation program finishes running, open a browser to log in to the following BIRT iHub consoles to perform user and administrator tasks:

- Information Console
  Perform tasks such as accessing folders and viewing designs and documents.
- Management Console
  Set up user accounts and schedule or run a design.

- Configuration Console

  Perform administrative operations, such as adding an Encyclopedia volume and making modifications to iHub parameters and server templates.

For more information on accessing BIRT iHub consoles, see Chapter 2, “Installing BIRT iHub,” earlier in this book.

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 Actuate 11 PostgreSQL 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 script-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 a new 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. Navigate to the following location:

      ```
      /home/actuate/AcServer/bin
      ```

   2. Execute the shutdown_srvr script:

      ```
      sh ./shutdown_srvr.sh
      ```

2. If the BIRT iServer Release 11 system uses the OOTB PostgreSQL RDBMS, shut down the Actuate 11 PostgreSQL for BIRT iServer service by executing the stopPostgreSQL.sh script:

   ```
   sh ./stopPostgreSQL.sh
   ```

   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.

3. 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 from 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. Back up 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. Navigate to the BIRT iHub AC_SERVER_HOME/bin folder. When installed in a typical side-by-side configuration, the BIRT iHub AC_SERVER_HOME refers to /home/actuate/iHub/AcServer.

   2. Execute the shutdown_srvr script:

      ```
      sh ./shutdown_srvr.sh
      ```

3. Shut down the Actuate PostgreSQL for BIRT iHub service by executing the stopPostgreSQL.sh script:

```
sh ./stopPostgreSQL.sh
```

4. 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 by performing the following tasks:

    1. Navigate to the BIRT iHub AC_DATA_HOME/config/iHub2 folder. When installed in a typical side-by-side configuration, the BIRT iHub AC_DATA_HOME refers to /home/actuate/iHub/AcServer/data.

    2. For example, copy acserverconfig.xml from the iServer AC_DATA_HOME/config/11SP4 folder to the BIRT iHub AC_DATA_HOME/config/iHub2 folder by executing the following command:

    ```
    cp /home/actuate/AcServer/data/config/11SP4
      /acserverconfig.xml .
    ```

    3. If migrating from a BIRT iServer Release 11 Service Pack 4 installation, copy the keys file from the iServer AC_DATA_HOME/config/11SP4 folder to the BIRT iHub AC_DATA_HOME/config/iHub2 folder by executing the following command:

    ```
    cp /home/actuate/AcServer/data/config/11SP4/keys .
    ```

5. 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 by performing the following tasks:

        1. Navigate to the BIRT iHub AC_DATA_HOME/encyc folder.

        2. Execute the following command:

        ```
        cp -r /home/actuate/AcServer/data/encyc/postgresql .
        ```

    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 = '/home/actuate/AcServer/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 = '/home/actuate/iHub/AcServer/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 = '/home/actuate/AcServer/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 = '/home/actuate/iHub/AcServer/data/encyc
   /postgresql/data'
```

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

6. Update the AC_DATA_HOME entry in the BIRT iHub acpmdconfig.xml file to use the value in the BIRT iServer acpmdconfig.xml file by performing the following tasks:

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

```
/home/actuate/AcServer/etc
```

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

```
<AC_DATA_HOME>/home/actuate/AcServer/data</AC_DATA_HOME>
```

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

```
/home/actuate/iHub/AcServer/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>/home/actuate/AcServer/data</AC_DATA_HOME>
   ...
```

7. If the BIRT iServer Release 11 system uses the out-of-the-box (OOTB) PostgreSQL RDBMS, start the Actuate PostgreSQL for BIRT iHub service by performing the following tasks:
   1. Navigate to the location of the script that starts the iHub PostgreSQL service. For example, in a default installation, navigate to the following location:

      ```
      /home/actuate/iHub/AcServer/bin
      ```

   2. Execute the following command:

      ```
      sh ./startPostgreSQL.sh
      ```

8. 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.
9. Restart BIRT iHub System by performing the following tasks:
   1. Navigate to the location of the script that starts the BIRT iHub System. For example, in a default installation, navigate to the following location:

      ```
      /home/actuate/iHub/AcServer/bin
      ```

   2. Execute the following command:

      ```
      sh ./start_srvr.sh
      ```

   3. 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. Add the paths to the Ant and Java executable programs to the PATH environment variable by performing the following tasks:
   1. Create an environment variable named ANT_HOME by executing the following command:

      ```
      export ANT_HOME=/home/actuate/iHub/AcServer/tools
         /apache-ant-1.8.2
      ```

   2. Create an environment variable named JAVA_HOME by executing the following command:

      ```
      export JAVA_HOME=/home/actuate/iHub/AcServer/jdk160
      ```

3. Add the folders containing the Ant and Java executable programs to the PATH variable by executing the following command:

   ```
   export PATH=$PATH:$ANT_HOME/bin:$JAVA_HOME/bin
   ```

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 Linux installation, the path is:

      ```
      /home/actuate/iHub/AcServer/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

      For example, in a default installation, edit the settings to point the following locations:

      ```
      AC_SERVER_HOME=/home/actuate/iHub/AcServer
      AC_DATA_HOME=/home/actuate/AcServer/data
      AC_CONFIG_HOME=/home/actuate/iHub/AcServer/data/config
         /iHub2
      ```

      Exit the editor, saving the file to disk.

2 Execute the following command to run 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
- Preparing to install an iHub cluster
- Performing a cluster installation using the wizard
- Adding a node to a cluster
- Finding the BIRT iHub home directory
- About the Java Development Kit

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:

- Shared configuration home and Encyclopedia volume directories
- Cluster resources, such as printers, database systems, and disk storage systems

It is the responsibility of the administrator performing the installation to make sure that all network sharing settings conform to the security policies in force for the environment.

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. It is the responsibility of the administrator performing the installation to make sure that all network sharing settings conform to the security policies in force for the environment.

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 to install an iHub cluster

When you create a BIRT iHub cluster, you must install and run all cluster nodes using the same administrative user account.

Creating an administrative user account

Before installing iHub, create a user account with the privileges to access the relevant files and directories. Like other Linux processes, the processes that perform BIRT iHub tasks run under a specific 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. If you exercise the same control over the operating system account for BIRT iHub that your site exercises for other system administrator and root accounts, you can maintain the same level of security.

Installation of the iHub under the root account is not recommended since the PostgreSQL server must start and be maintained under an unprivileged user ID to prevent system security compromise. If installed under the root account, the default installation will be unable to set up the PostgreSQL schema and Actuate sample Encyclopedia.

Installing X-frame buffer

Xvfb is an X-Windows server that has neither a graphics card nor a physical graphics display. iHub uses the X-Server for font-rendering information and to generate graphics in documents. Normally, an X-Server requires a graphics card and physical graphics display on the BIRT iHub machine, but you can use Xvfb in place of these components.

The Xvfb software installed with iHub includes Type 1 fonts. Actuate maps these fonts to Microsoft Windows fonts for consistent graphics rendering on the various platforms.

The Xvfb software requires:

- X libraries installed on the iHub machine
  If you choose to install Xvfb, the installation script searches for the required libraries and displays a message if the install script cannot find the required libraries on the machine.
- Variables set to the path of the Xvfb libraries
  - XVFBDISPLAY variable in start_srvr.sh.
  - display_value in pmd11.sh.

To view and print the documents from iHub, you need to set these variables only if you install Xvfb software yourself.

The DISPLAY environment variable specifies the X-Windows server used by the BIRT iHub machine. For example, if the BIRT iHub machine is running X-Windows, it sets DISPLAY to the local machine:

```
# setenv DISPLAY :0.0
```

If you use a separate machine as the X-Windows server, specify the machine name in the environment variable DISPLAY. The following example sets DISPLAY to use an X-Windows server on a machine named urup:

```
# setenv DISPLAY urup:0.0
```

The original source code for Xvfb is included as a component of X11R6, but not in earlier X-Window system releases.

Performing a cluster installation using the wizard

The following section describes how to install an iHub cluster node in the Linux operating system using a Linux system as the example.

How to install a cluster node in Linux

1. Download the required files from the FTP software distribution site. Extract the files.
2. To install the server files, execute the isinstall script:

   ```
   sh ./isinstall.sh
   ```

   The script displays a series of prompts. Respond to the prompts as described in the following procedures.
3. The license agreement appears, as shown in Figure 5-1.

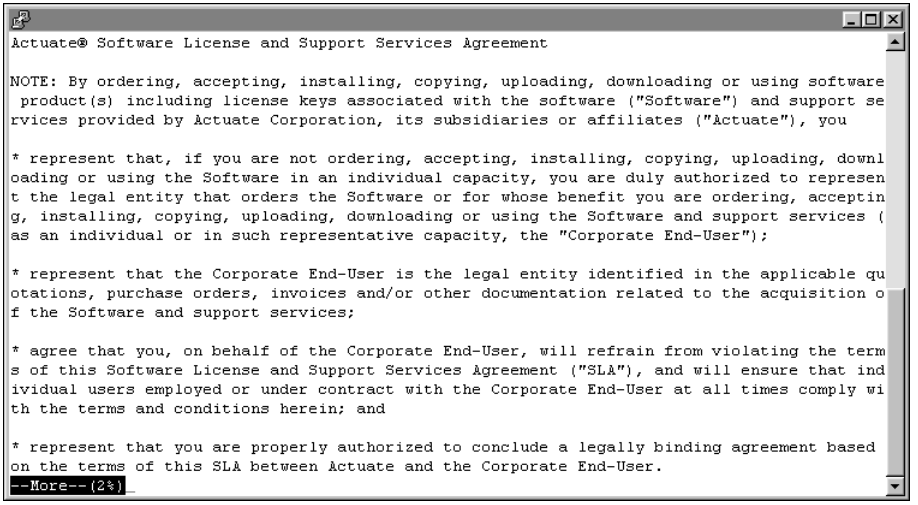


Figure 5-1 Reviewing the license agreement

4. Read the license agreement and press Enter to continue the installation. At the prompt, type y for yes if you accept the licensing terms, as shown in Figure 5-2.

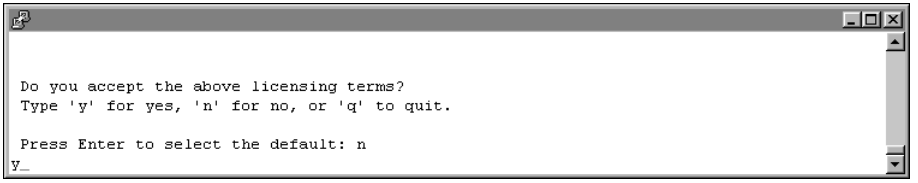


Figure 5-2 Accepting the licensing terms

5 The introduction to the installation appears, as shown in Figure 5-3.

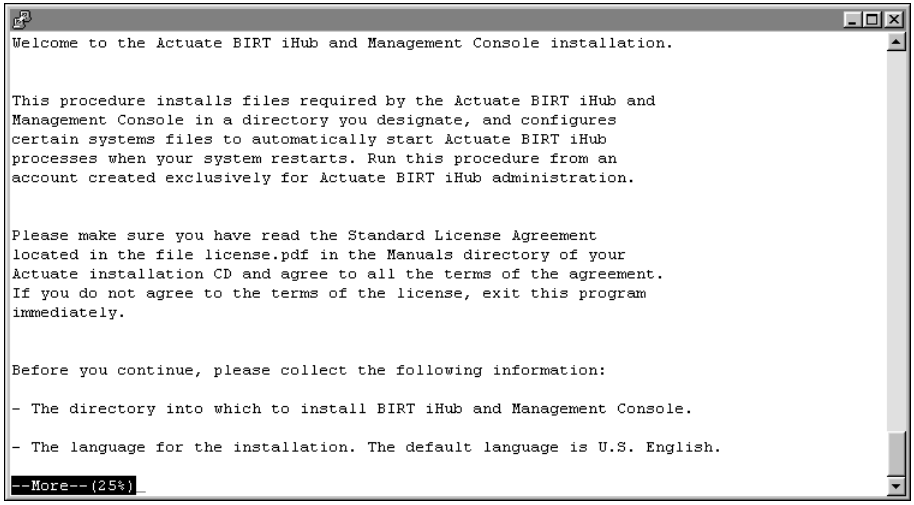


Figure 5-3 Reviewing the introductory information

6 Press Enter after reviewing the introductory information, as shown in Figure 5-4.

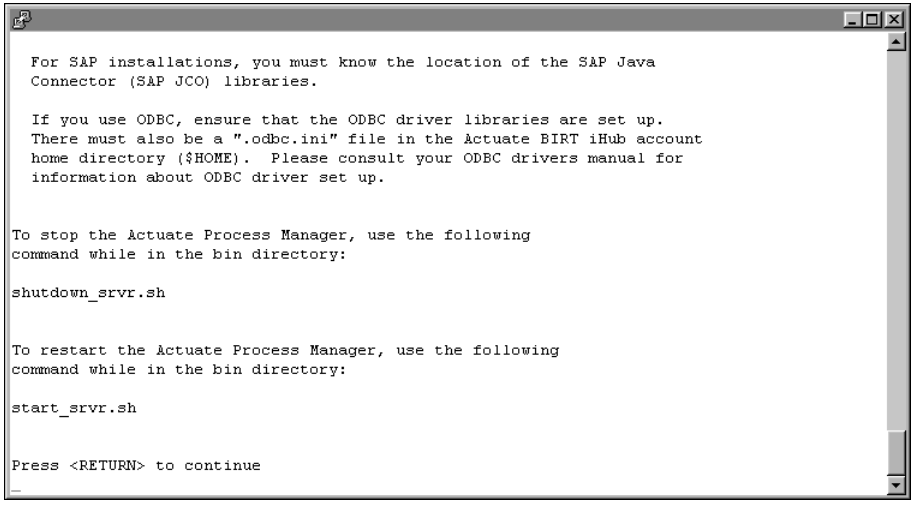


Figure 5-4 Finishing the review of introductory information

7 Press Enter to accept the default location for the installation binaries, as shown in Figure 5-5. Alternatively, type a different directory and press Enter.

The installation program creates the AcServer directory in your chosen location and installs the files.

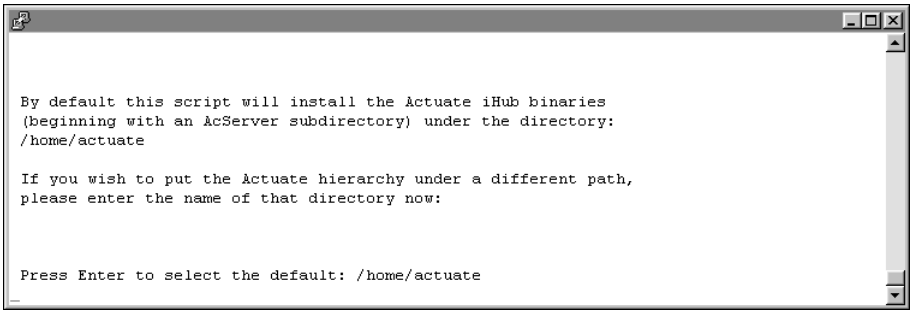


Figure 5-5 Specifying the installation directory

8 Press Enter to accept the default installation directory, AC_SERVER_HOME /data, for the iHub data as shown in Figure 5-6. Alternatively, choose a different directory.

iHub uses this data location to store the iHub Encyclopedia volume data, including PostgreSQL metadata, logs, and other files. The default path is AC_SERVER_HOME/data, which is referred to in the iHub documentation by the environment variable AC_DATA_HOME.

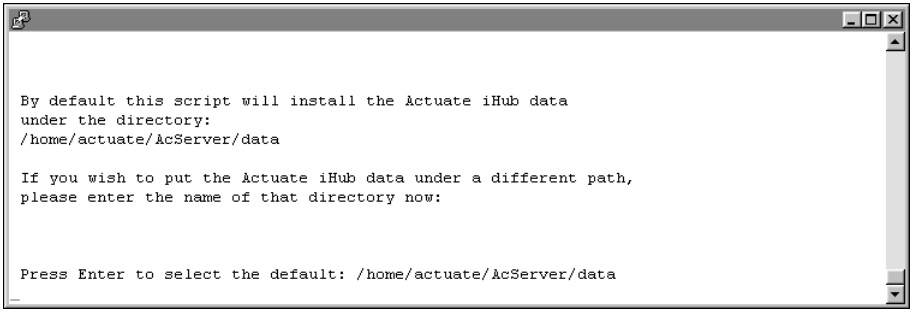


Figure 5-6 Specifying the data installation directory

9 Press Enter to accept the default option of creating the directory for data, as shown in Figure 5-7. Alternatively, type n for no, or q to quit, and press Enter.

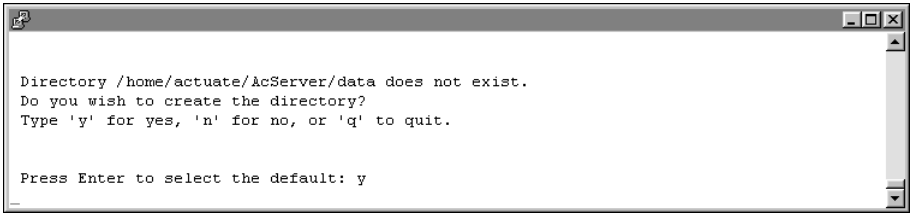


Figure 5-7 Creating the AC_DATA_HOME directory

10 The installer copies prerequisite files to the destination directory, as shown in Figure 5-8. After copying the prerequisite files, the installation continues.

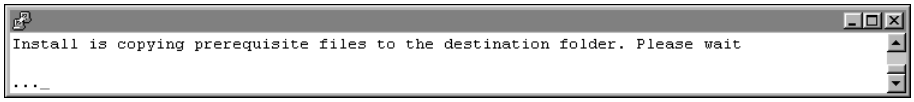


Figure 5-8 Copying prerequisite files

11 Press Enter to choose the default option, Server with Management Console, as shown in Figure 5-9. A cluster node must have access to Configuration Console. Configuration Console installs with Management Console.

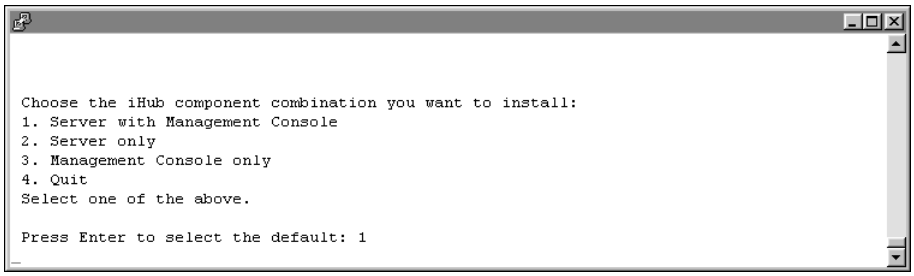


Figure 5-9 Choosing the components to install

12 Type 1 and press Enter to select Cluster Server for installation, as shown in Figure 5-10. Alternatively, choose a different type of iHub to install.

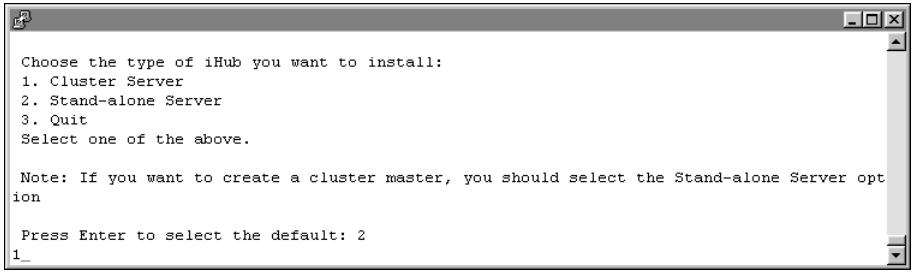


Figure 5-10 Specifying the type of iHub to install

13 Specify the path to the configuration home location, and press Enter, as shown in Figure 5-11.

The configuration home location is the shared directory of the cluster containing the files, acserverconfig.xml, acserverlicense.xml, and acconfigowner.lock. In an iHub installation, the configuration files are located in AC_DATA_HOME/config/iHub2 by default. For more information about configuring network sharing, see "Adding a node to a cluster," later in this chapter.

Figure 5-11 Specifying the configuration home location

14 If the specified location for the configuration home directory does not yet exist, you are prompted to create the directory. Press Enter to accept the default option which creates the directory, as shown in Figure 5-12. Alternatively, press n for no, or q to quit.

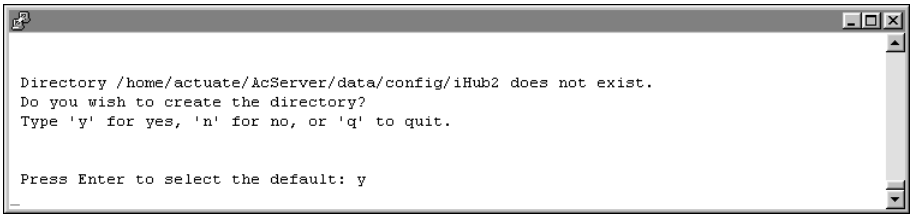


Figure 5-12 Creating the configuration home location

15 If you chose to install Server with Management Console instead of Server only, perform the following steps:

1. Press Enter to select the default locale, which is English (United States), as shown in Figure 5-13. Alternatively, select a different locale. If you do not see the locale for your region, type m for more and press Enter.

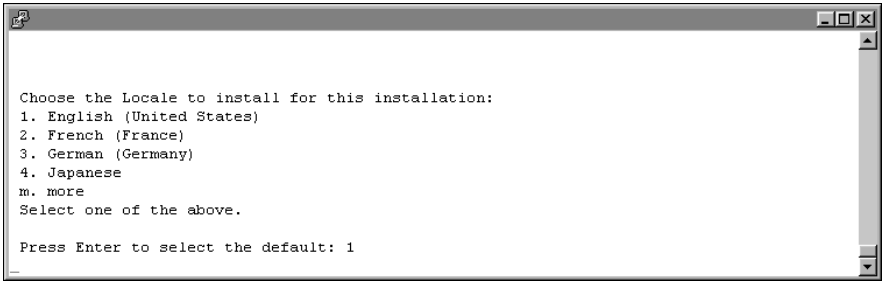


Figure 5-13 Specifying a locale

2. Press Enter to select the default time zone, which is America/Los_Angeles as shown in Figure 5-14. Alternatively, select another time zone from the numbered list.

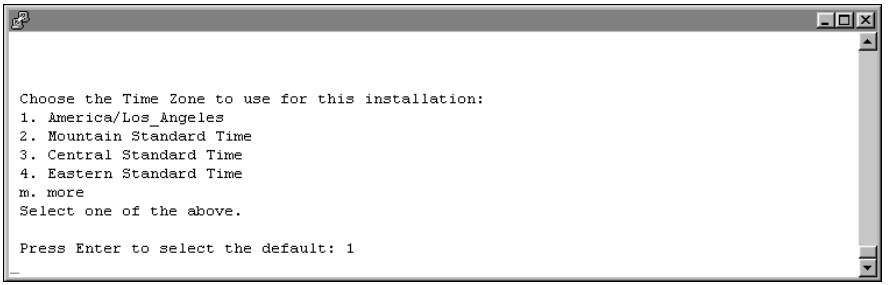


Figure 5-14 Specifying a time zone

16 To evaluate the product using the included evaluation software license, press Enter, as shown in Figure 5-15. Alternatively, type 2, then type the path to the license file that you purchased.

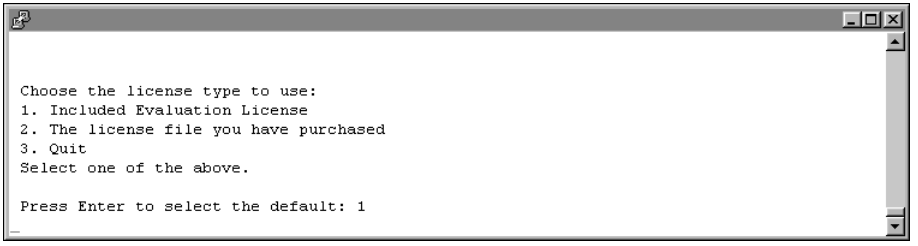


Figure 5-15 Specifying license type

17 Press Enter to select the default host name, the name of your machine, where the Process Management Daemon (PMD) runs, as shown in Figure 5-16. Alternatively, type a different IP address or hostname.

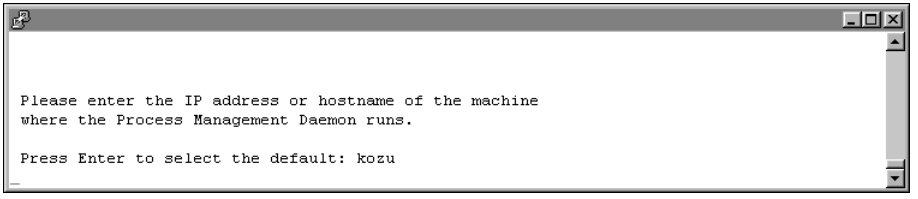


Figure 5-16 Specifying the machine on which the PMD runs

18 Press Enter to accept the default port number where the Process Management Daemon (PMD) listens for requests, as shown in Figure 5-17. Alternatively, type a different port number.

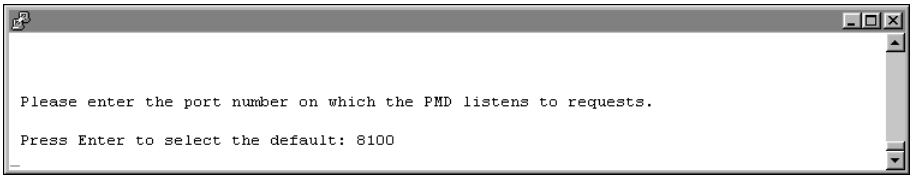


Figure 5-17 Specifying the port number on which the PMD listens

19 Specify the administrator password, as shown in Figure 5-18.

You use this password to log in to the iHub Configuration Console.

Figure 5-18 Specifying the administrator password

20 Re-enter the password of the administrator, as shown in Figure 5-19.

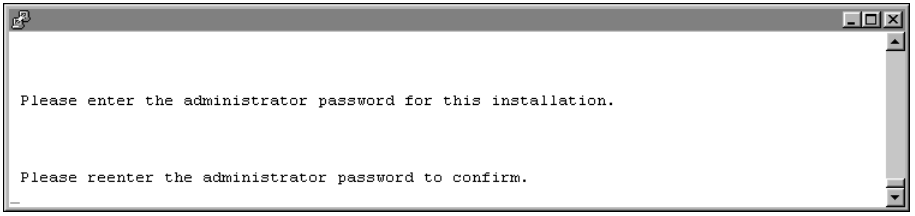


Figure 5-19 Re-entering the administrator password

21 Press Enter to accept the default option of not using any database drivers/clients, as shown in Figure 5-20. Alternatively type y for yes, specify the database drivers/clients you wish to use and press Enter.

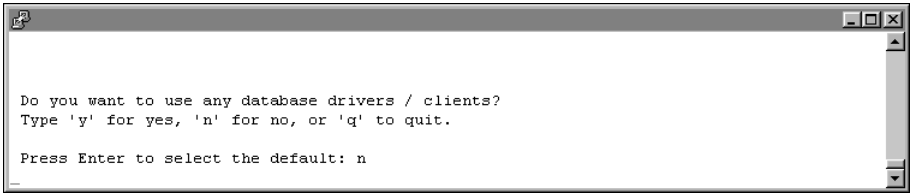


Figure 5-20 Specifying whether to use database drivers/clients

22 Specify what kind of X-Server you want to use, if any. To accept the default, press Enter, as shown in Figure 5-21. For more information about installing X-server, see “Installing X-frame buffer,” earlier in this chapter.

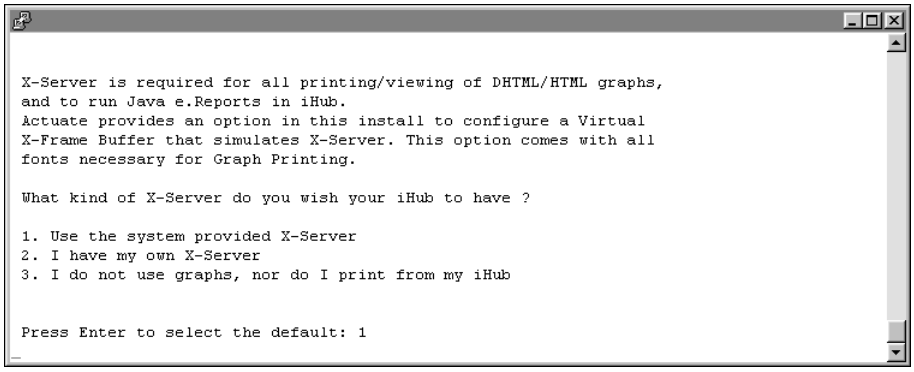


Figure 5-21 Specifying what kind of X-Server to use, if any

23 If you chose to install Server with Management Console instead of Server only, perform the following steps:

1. Press Enter to accept the default hostname, the name of your machine, that Management Console uses to contact the Process Management Daemon (PMD), as shown in Figure 5-22. Alternatively, type a different IP address.

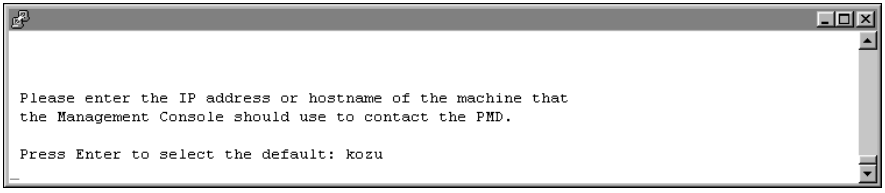


Figure 5-22 Specifying the machine host name that Management Console uses to contact the PMD

2. Press Enter to accept the default port number, 8100, on which the Process Management Daemon (PMD) listens for requests from Management Console, as shown in Figure 5-23. Alternatively, type a different port number.

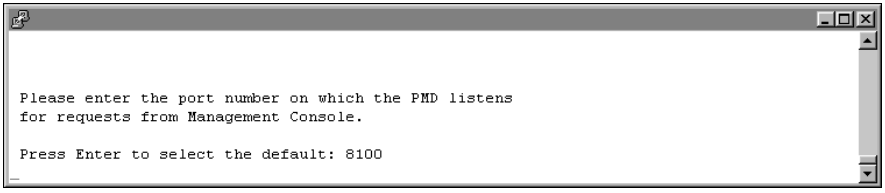


Figure 5-23 Specifying the port number for the PMD to listen for requests from Management Console

3. Press Enter to accept the default hostname, the name of your machine, as shown in Figure 5-24. Alternatively, type a different IP address.

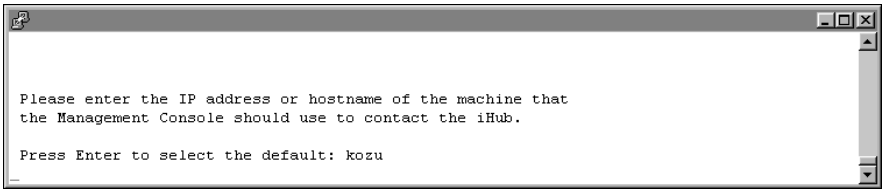


Figure 5-24 Specifying the name Management Console uses to contact iHub

4 Press Enter to accept the default port number, 8000, as shown in Figure 5-25. Alternatively, type a different port number.

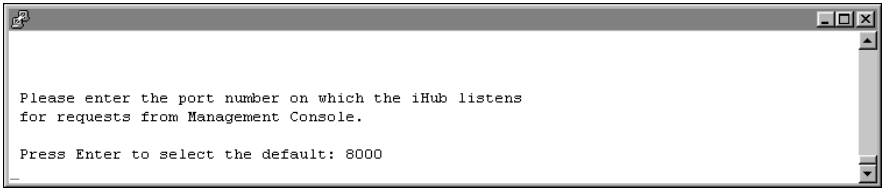


Figure 5-25 Specifying the port number for listening for requests from Management Console

5 Press Enter to accept the default name, the name of your machine, for the Encyclopedia volume to use with Management Console, as shown in Figure 5-26. Alternatively, type a different name for the Encyclopedia volume.

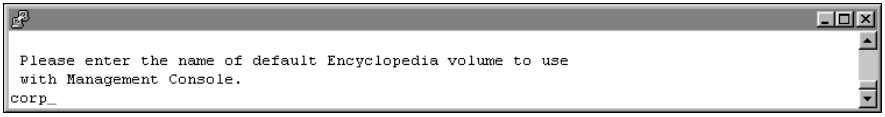


Figure 5-26 Specifying the name of the default Encyclopedia volume

6 Press Enter to accept the default name, acadmin, for the HTTP server context root, as shown in Figure 5-27. Alternatively, type a different name.

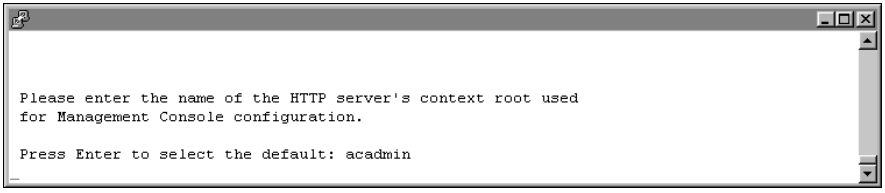


Figure 5-27 Specifying the name of the HTTP server context root

24 Press Enter to accept the default port number, 8900, on which the application container listens for requests, as shown in Figure 5-28. Alternatively, choose a different port.

You connect to the port from your browser when accessing various features of iHub.

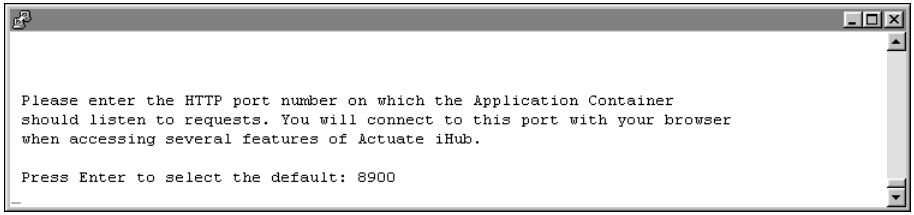


Figure 5-28 Specifying the application container listening port number

25 Review the settings, as shown in Figure 5-29, then specify whether you accept the settings. Press Enter to accept the default, y for yes. Alternatively, type n for no, or q to quit.

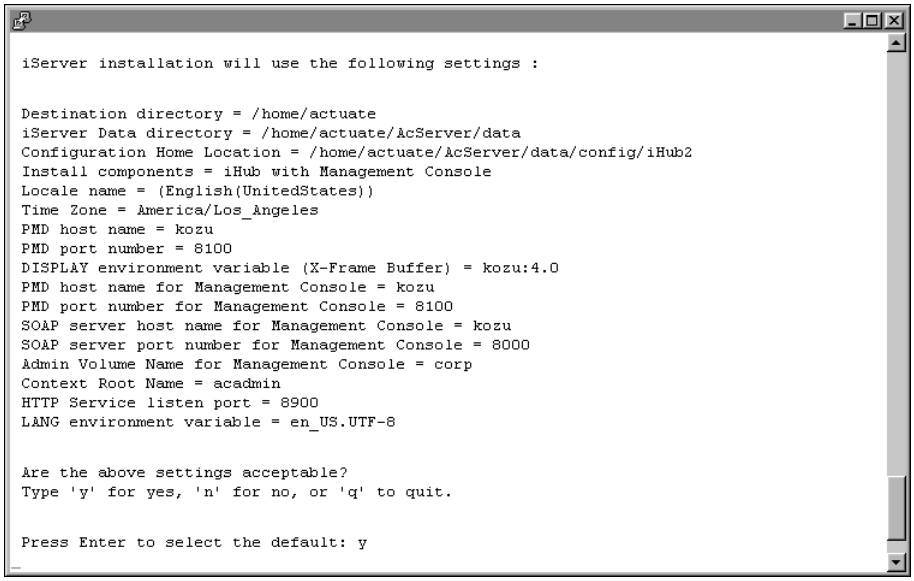


Figure 5-29 Reviewing settings for a Server with Management Console install

26 The installation program installs iHub, and displays an indicator showing how the installation is progressing, as shown in Figure 5-30.

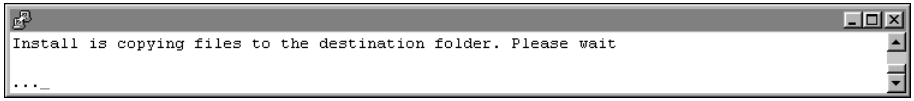


Figure 5-30 Copying iHub files to your destination folder

27 At the end of the installation, the program asks if you want to start iHub. Accept the default, y for yes, to start the Process Management Daemon (PMD), as shown in Figure 5-31.

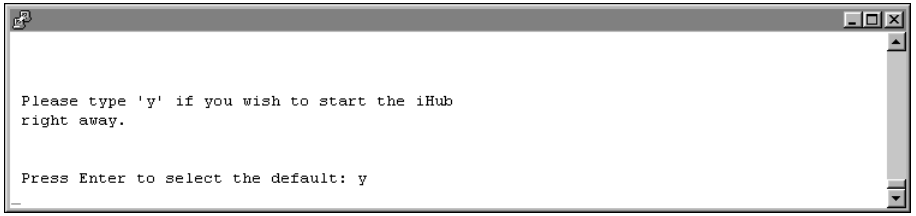


Figure 5-31 Specifying whether to start iHub

28 When the installation program finishes, it provides additional information about localization, logging in using an account with root permissions to start iHub, and installing online help and manuals, as shown in Figure 5-32.

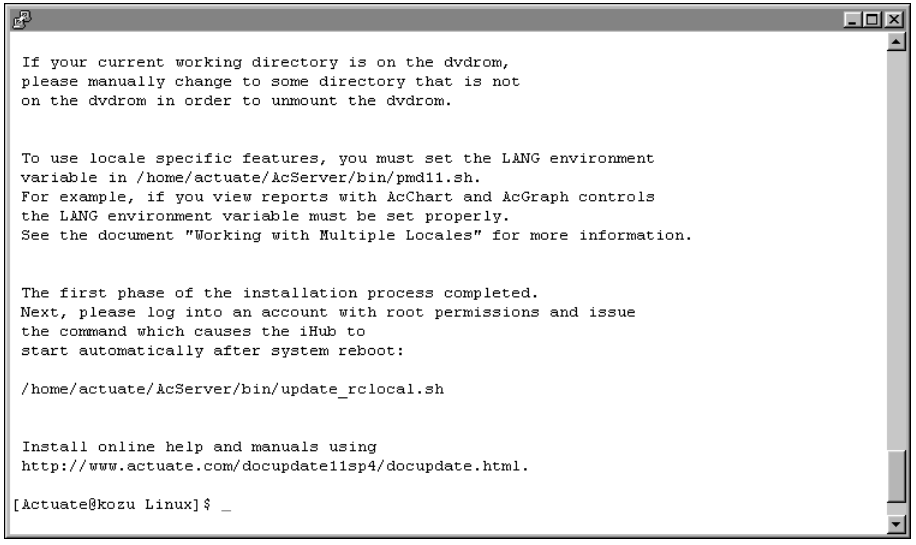


Figure 5-32 Viewing information about localization, logging in, and installing online help

Adding a node to a cluster

After installing a node on a machine, the administrator must still configure sharing and add the node to the cluster. When adding a node to a cluster setup, the administrator must verify that the configuration home directory specified during the install procedure points to the shared configuration home directory and all Encyclopedia volume resources are accessible.

The following section refers to the machine containing the shared configuration directory as node1 and the cluster node accessing these shared resources as node2.The following example assumes that both the configuration folder and Encyclopedia volume folders are located on node1, although in a more complex installation, these configuration and volume resources may reside in another network location.

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

- On node1, the Administrator shares the configuration folder and any Encyclopedia volume folders that a cluster node accesses.
- On node2, the Administrator:
  - Creates folders on which to mount the node1 shared folders
  - Creates a mapping between the node1 and node2 shared folders
  - Mounts the node1 shared folders on the node2 machine

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.

The following instructions provide a basic reference example of the operations required to configure folder sharing in a Linux environment that supports using the Network File System (NFS), a common, standard, distributed file system protocol.

How to share the configuration and Encyclopedia volume files and folders

In a default iHub installation, a cluster node requires shared, read-write access to the following system resources:

- AC_DATA_HOME/config/iHub2
  In an iHub installation, the configuration files are located in AC_DATA_HOME/config/iHub2.
- AC_DATA_HOME/encyc or other volumes, including all file, fileType, status, and tempRov subfolders
  In an iHub installation, where there has been no activity on the system, the status or tempRov folders may not exist. These folders contain information about job details and completion notices and do not appear until a job executes.

To give a cluster node read-write access to these files and folders, perform the following tasks:

1 Log in to node1 as the root user.

2 Add the following entries to the /etc/exports file:

```
/home/actuate/AcServer/data/config/iHub2
   *(rw,fsid=1,no_root_squash)
/home/actuate/AcServer/data/encyc
   *(rw,fsid=2,no_root_squash)
```

3 Start the NFS server processes by executing the following command:

```
service nfs restart
```

4 Log in to node2 as the actuate user.

5 Create the following directory paths:

```
/home/actuate/AcServer/data/config/iHub2
/home/actuate/AcServer/data/encyc
```

6 Log off node2.

7 Log in to node2 as the root user.

8 Add the following entries to the /etc/fstab file:

```
<node1 hostname>:/home/actuate/AcServer/data/config/iHub2
   /home/actuate/AcServer/data/config/iHub2 nfs nfsvers=3 0 0
<node1 hostname>:/home/actuate/AcServer/data/encyc /home/
   actuate/AcServer/data/encyc nfs nfsvers=3 0 0
```

9 Mount the shared folders on node1 by executing the following commands:

```
mount /home/actuate/AcServer/data/config/iHub2
mount /home/actuate/AcServer/data/encyc
```

The administrator must also verify or edit the shared acpmdconfig.xml file to contain the following information:

- <AC_CONFIG_HOME> to point to the shared configuration home directory for the cluster
- <AC_TEMPLATE_NAME> to specify the server template from the available server templates listed in the shared acserverconfig.xml file

How to verify and edit acpmdconfig.xml file settings

To verify and edit acpmdconfig.xml file settings, perform the following tasks:

1 Shut down the recently installed cluster node.

2 Using a text editor, open acpmdconfig.xml, which by default is located in AC_SERVER_HOME/etc.

3 Verify or edit <AC_CONFIG_HOME> to point to the shared configuration home directory for the cluster, as shown in the following code:

```
<AC_CONFIG_HOME>/home/actuate/AcServer/data/config/iHub2
   </AC_CONFIG_HOME>
```

This location is the path that you specified for the configuration home directory during the install procedure.

4 Verify or edit <AC_TEMPLATE_NAME> to specify the server template name from the available server templates listed in the shared acserverconfig.xml file, as shown in the following code:

```
<AC_TEMPLATE_NAME>urup</AC_TEMPLATE_NAME>
```

In the example, urup is server template name

5 Save acpmdconfig.xml.

The administrator must also verify or edit the shared acserverconfig.xml file to contain the following information:

- <ServerFileSystemSetting> points to the shared drive location that contains the Encyclopedia volume data files.
- server <ConnectionProperty> specifies the network name of the node that contains the shared Encyclopedia volume database.

How to verify and edit acserverconfig.xml file settings

To verify and edit acserverconfig.xml file settings, perform the following tasks:

1 Stop the Actuate BIRT iHub service running on the node that contains the shared configuration home directory.

2 Using a text editor, open the acserverconfig.xml file in the configuration home directory.

In an iHub installation, the configuration files are located in AC_DATA_HOME/config/iHub2 by default. The location is the path that you specified for the configuration home directory during the install procedure.

3 In <Template> settings for the node, verify or edit <ServerFileSystemSettings> to make sure the path <ServerFileSystemSetting> points to the location that contains the Encyclopedia data files, by performing the following tasks:

1 Locate the <ServerFileSystemSettings> element under the <Template> element.

2 In <ServerFileSystemSetting>, locate:

```
<ServerFileSystemSettings>
   <ServerFileSystemSetting
      Name="DefaultPartition"
      Path="$AC_DATA_HOME$/encyc"/>
</ServerFileSystemSettings>
```

3 Change Path from the AC_DATA_HOME variable notation to the full path specification, as shown in the following code:

```
<ServerFileSystemSettings>
   <ServerFileSystemSetting
      Name="DefaultPartition"
      Path="/home/actuate/AcServer/data/encyc"/>
</ServerFileSystemSettings>
```

The Path setting for DefaultPartition is /home/actuate/AcServer/data/encyc. Do not use the AC_DATA_HOME variable notation.

4 In <MetadataDatabase> settings, verify or edit the <ConnectionProperty> for the server to make sure that it specifies the network name, not localhost, of the node on which the Encyclopedia volume database resides, by performing the following tasks:

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

2 In <ConnectionProperties> locate:

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

3 Change Value from localhost to the name of the machine on which the Encyclopedia volume database resides, such as urup, as shown in the following code:

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

5 Save acserverconfig.xml.

Start Actuate BIRT iHub on each cluster node. The new cluster node will automatically read the settings in the acserverconfig.xml file in the shared configuration directory to access its template, then join the cluster.

How to start an iHub cluster using Configuration Console

To start iHub using Configuration Console manually, perform the following tasks:

1 On the node containing the configuration home directory for the cluster, log in to Configuration Console and choose Advanced view. Choose Servers, then choose Start New Server.

2 On Servers—Start New Server, as shown in Figure 5-33, perform the following tasks:

1 In Server name, type the name of the cluster node.

2 In Host Name or IP Address, type the name or IP address of the cluster node.

3 In iHub Process Manager Port Number, type the Daemon listen port number. The default value for this port is 8100. You specify this port number during the install procedure.

4 In Server template name, choose the name of the template that the cluster node uses.

Choose OK.

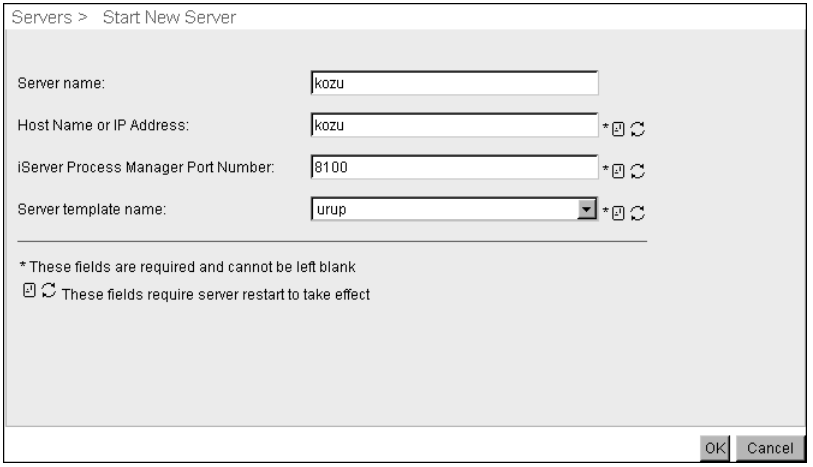


Figure 5-33 Preparing to start a new server

3 Log out of Configuration Console.

4 Restart the Actuate BIRT iHub services on the node containing the configuration home directory for the cluster then the new node.

5 Log in to Configuration Console and choose Advanced view. Choose Servers from the side menu. The new cluster node automatically reads the acserverconfig.xml in the shared configuration home directory to access its template, then joins the cluster.

Finding the BIRT iHub home directory

The environment variable for the iHub home directory is AC_SERVER_HOME. The iHub installation program sets the variable to the path of your iHub login environment.

How to find the home directory for BIRT iHub on a Linux system

If you use the C Shell on a Linux system, you can look in your .cshrc file to see the value of AC_SERVER_HOME:

```
$ setenv AC_SERVER_HOME /usr/local/AcServer
```

If an iHub process is running on the system, you can also use the following ps command piped to the grep command to find the pmd11 executable, which runs from the iHub home bin directory:

```
# ps -ef | grep pmd
```

About the Java Development Kit

The BIRT iHub installation routine installs the Java SDK files under the directory specified in the environment variable AC_SERVER_HOME:

```
AC_SERVER_HOME/jdk160
```

Some operating systems require an operating system upgrade or patch to use JRE 6.0. For information about requirements for your operating system, see your operating system documentation. Also, see the Actuate Support Lifecycle Policy and Supported Products Matrix on the Actuate Support web site. You can access the Support site at the following URL:

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

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

- AC_JAVA_HOME
- AC_JVM_HOME
- AC_JRE_HOME in the PMD startup script, pmd11.sh
- 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 Actuate charts incorrectly.

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 and configuring the appropriate environment variables, 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, iHub 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 TAR 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 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
```

Set the JDK_HOME environment variable to point to the location of the JDK.

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 Linux setup performed using the wizard-based install program, the iHub run-time environment installs in the following directory:

```
$HOME/AcServer
```

The data installs in the following directory:

```
$HOME/AcServer/data
```

In a cloud deployment, Actuate recommends installing in an alternative directory. For example, in Linux, install the run-time environment in the following directory:

```
/home/actuate/Actuate/AcServer
```

The data installs in the following directory:

```
/home/actuate/Actuate/AcServer/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 $HOME/AcServer, such as $HOME/Actuate.
2. Extract the contents of ActuateBIRTiHub.tar.gz to the folder created in the previous step.
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.sh script automatically detects the location of AC_SERVER_HOME and AC_JAVA_HOME in most cases. 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 and set the values for these properties manually.

   Specify the location of AC_SERVER_HOME and AC_JAVA_HOME in the setupiHub script by performing the following tasks:

   1. Using a text editor, open the script, setupiHub.sh, 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 script

```
#!/bin/sh
# Script that sets up the iHub and starts the iHub

#Title Actuate BIRT iHub 2 setup and start script for
   evaluation

AC_SERVER_HOME=/home/actuate/Actuate/AcServer
AC_JAVA_HOME=/home/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 in to 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. Navigate to the folder where you extracted the iHub package.
2. Execute setupiHub.sh by entering the following command:

   ```
   sh ./setupiHub.sh
   ```
3. The script prompts you to choose one of the following stand-alone or cluster options in setting up iHub, as shown in Figure 6-1:
   - Stand-alone
     - 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.

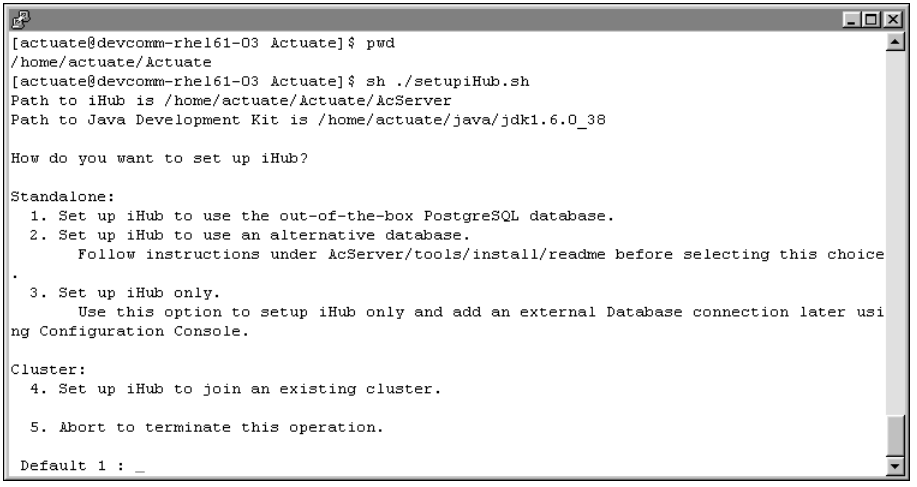


Figure 6-1 Choosing setup type

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.tar.gz 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 6-1.

The script performs the following tasks, as shown in Figure 6-2:

- 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

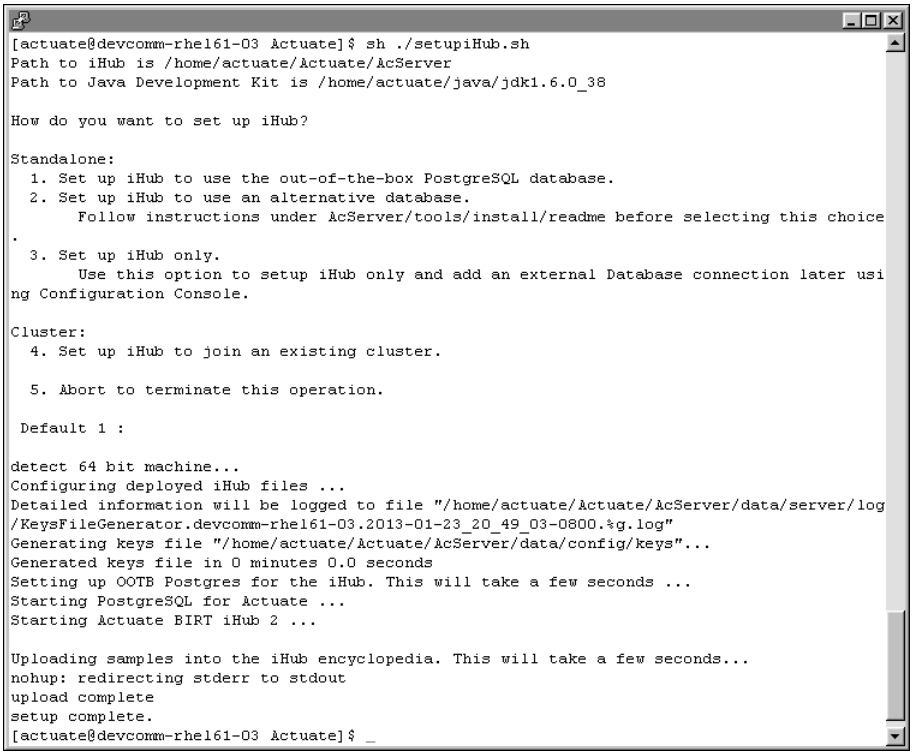


Figure 6-2 Completing the iHub setup

Accessing Information, Management, and Configuration Consoles

After the script finishes running, open a browser to log in to the following BIRT iHub consoles to perform user and administrator tasks:

- Information Console
  Perform tasks such as accessing folders and viewing designs and documents.

  To access Information Console, open a browser manually and enter the following URL, as shown in Figure 6-3:

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

Figure 6-3 Viewing Welcome to Actuate Information Console

- Management Console
  Set up user accounts and schedule or run a design.

  To access Management Console, open a browser manually and enter the following URL, as shown in Figure 6-4:

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

Figure 6-4 Logging in to Management Console

- Configuration Console
  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 6-5:

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

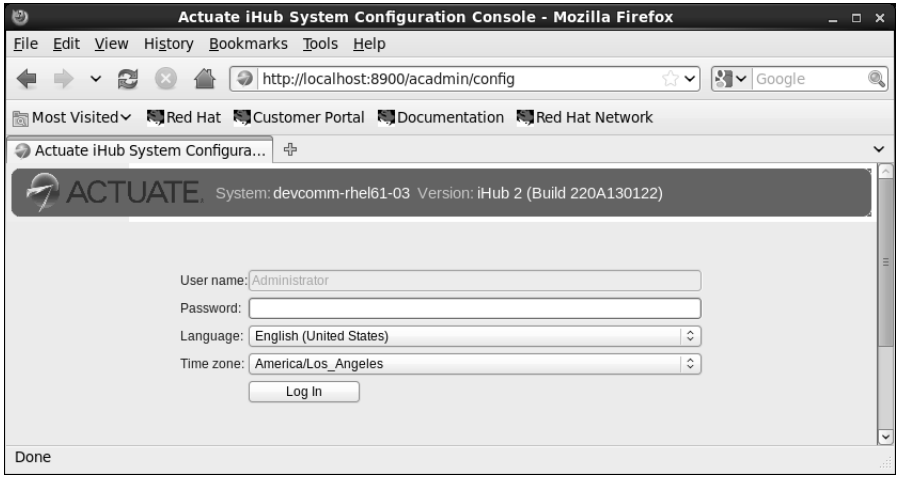


Figure 6-5 Logging in to Configuration Console

When starting PostgreSQL for BIRT 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 BIRT iHub

To stop 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:

```
sh ./shutdown_srvr.sh
```

To start 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:

```
sh ./startsrvr.sh
```

How to stop and start PostgreSQL for Actuate iHub

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:

```
sh ./stoppostgresql.sh
```

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:

```
sh ./startpostgresql.sh
```

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.tar.gz 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. In a command prompt, navigate to AC_SERVER_HOME\tools\install. For example:

   ```
   /home/actuate/Actuate/AcServer/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 /home/actuate/Actuate/AcServer.
- 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: /home/actuate/Actuate/
   AcServer
AC_SERVER_HOME=/home/actuate/Actuate/AcServer

#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.

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

- Sets up the iHub deployment files, including log and security keys files
- Sets up and starts BIRT iHub 2
- Creates the Encyclopedia volume
- Uploads the Encyclopedia volume sample content

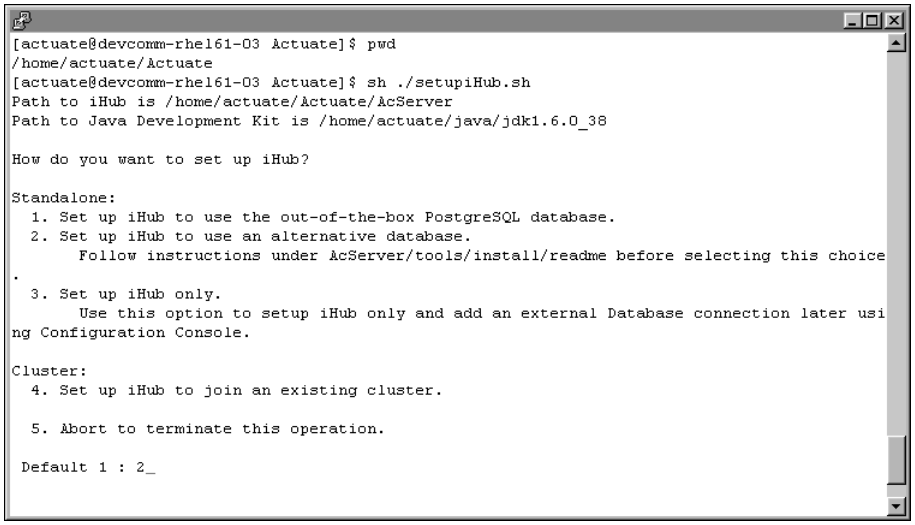


Figure 6-6 Completing the iHub alternative database setup

After the script finishes running, open a browser manually and enter the following URL, as shown in Figure 6-7:

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

Log in to Information Console to perform tasks such as accessing folders and viewing designs and documents.

Figure 6-7 Viewing Welcome to Actuate BIRT iHub

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
  - Data partition and Encyclopedia volume

Setting up iHub only using the cloud deployment package

After extracting the contents of ActuateBIRTiHub.tar.gz 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 6-8:

   - Sets up the iHub deployment files, including log and security keys files
   - Sets up and starts BIRT iHub 2

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

   1. Navigate to AC_SERVER_HOME/bin.
   2. Type the following command and press Enter, as shown in Figure 6-9:

      ```
      sh ./startsrvr.sh
      ```

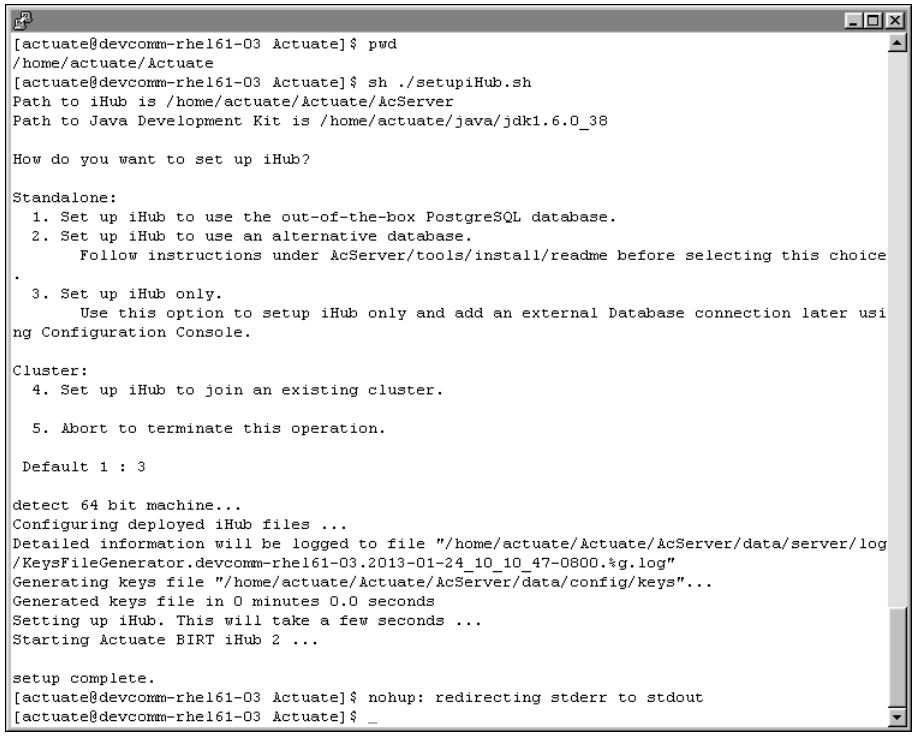


Figure 6-8 Completing the iHub only setup

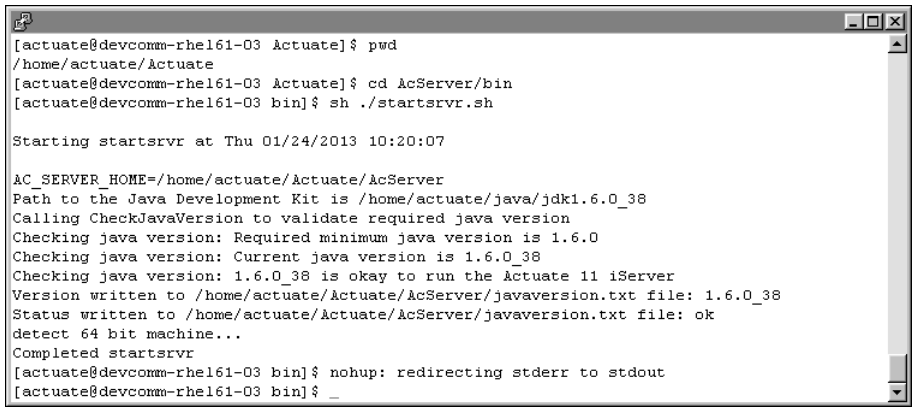


Figure 6-9 Starting Actuate BIRT iHub

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 role
- 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 PostgreSQL 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 6-10. The administrator does not have a password yet, so leave Password blank.

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

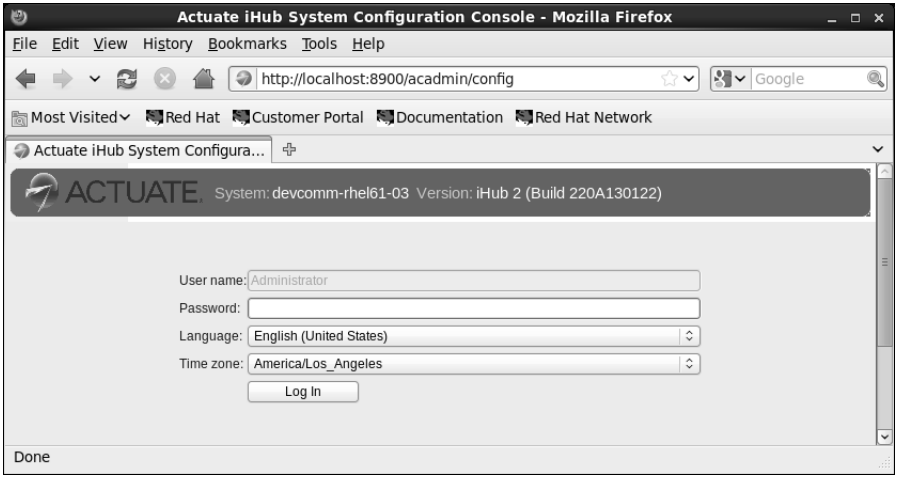


Figure 6-10 Logging in to Configuration Console

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 6-11.

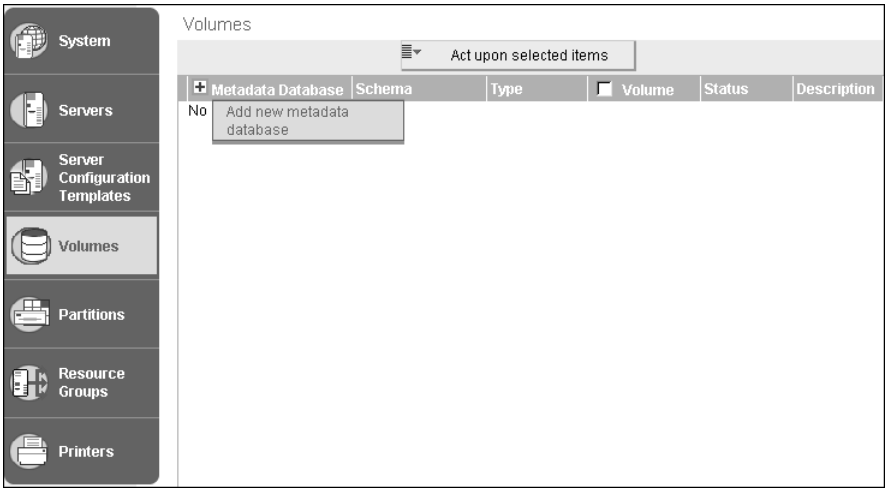


Figure 6-11 Adding a new metadata database

If a warning message appears, stating that Actuate supports using only one metadata database and one volume schema at a time, choose OK.

2 On New Metadata Database, perform the following tasks, as shown in Figure 6-12:

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.

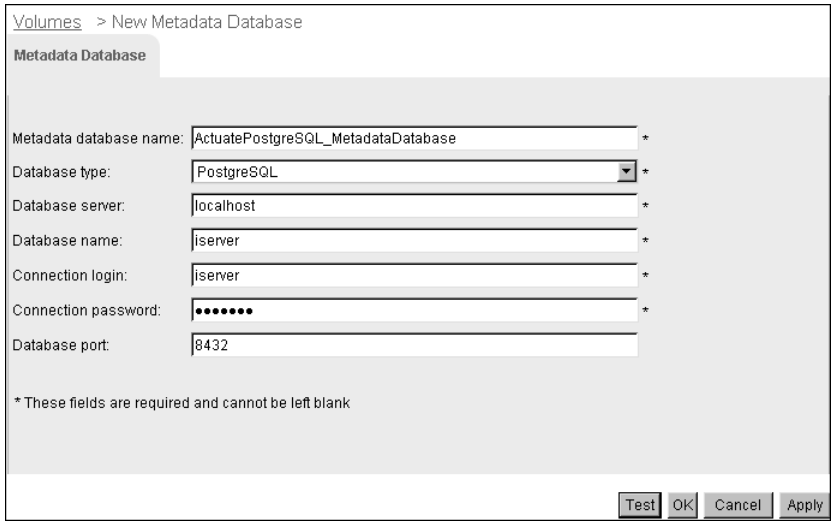


Figure 6-12 Adding new metadata database properties

8 Choose Test.

The message that the connection tested successfully appears, if the connection properties are correct and there are no problems communicating with the database system, as shown in Figure 6-13. Choose OK.

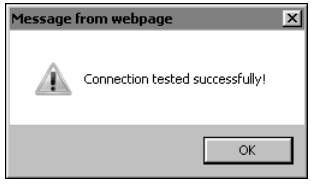


Figure 6-13 Viewing the connection tested successfully message

On New Metadata Database, choose OK.

9 On Volumes, the ActuatePostgreSQL_MetadataDatabase appears, as shown in Figure 6-14.

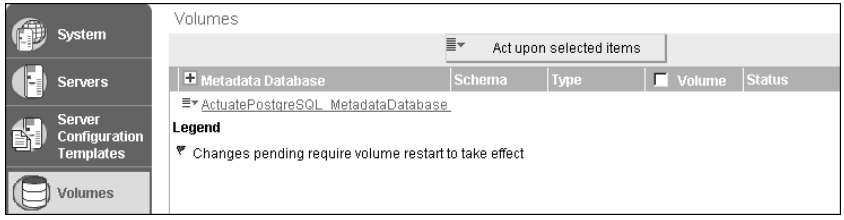


Figure 6-14 Viewing the metadata database

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 6-15.

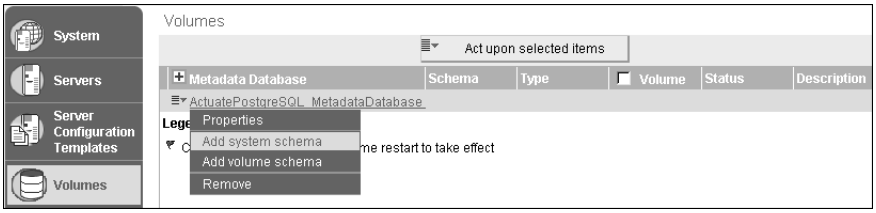


Figure 6-15 Choosing Add system schema

2 On New System Schema, perform the following tasks, as shown in Figure 6-16:

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.

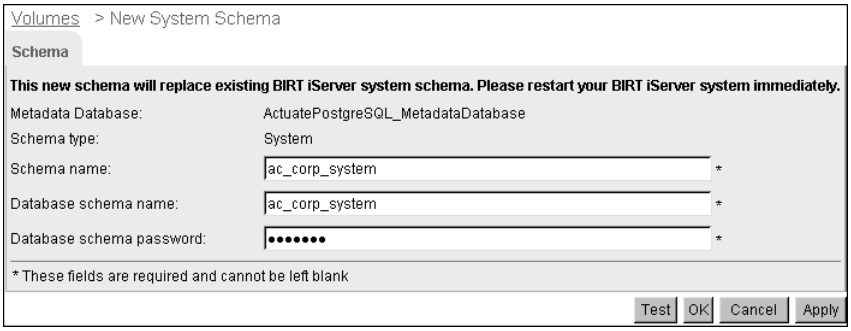


Figure 6-16 Adding a new system schema

4 Choose Test. If successful, 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 6-17.

Figure 6-17 Viewing message to restart iHub system

Choose OK.

5 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 6-18.

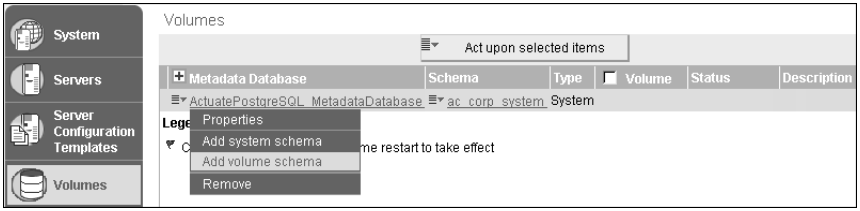


Figure 6-18 Choosing Add volume schema

2 On New Volume Schema, as shown in Figure 6-19, 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.

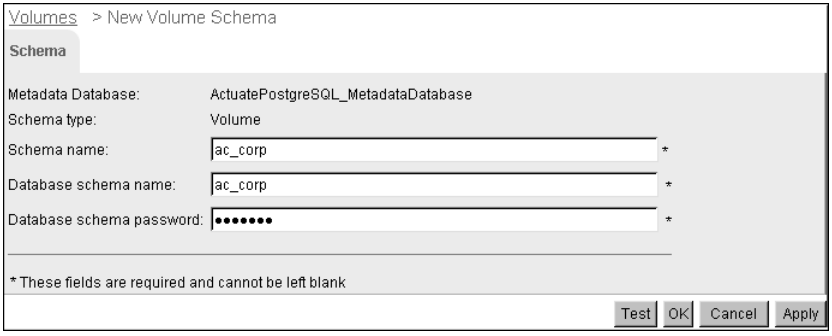


Figure 6-19 Adding a new Encyclopedia volume schema

4 Choose Test.

The message that the connection tested successfully appears, if the connection properties are correct and there are no problems communicating with the database system, as shown in Figure 6-20. Choose OK.

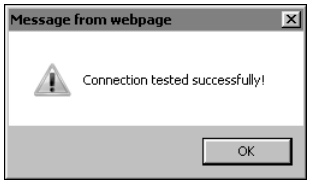


Figure 6-20 Viewing the connection tested successfully message

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 6-21.

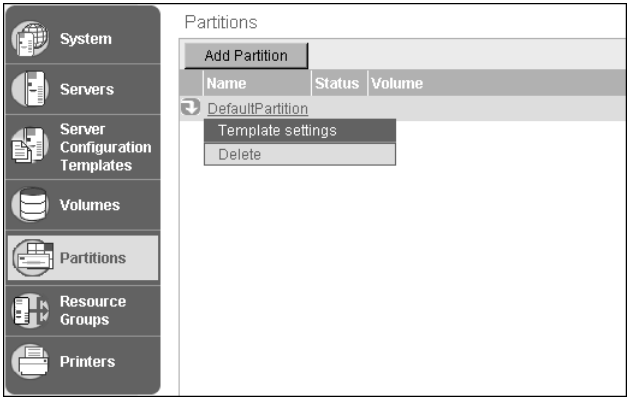


Figure 6-21 Deleting DefaultPartition

Choose OK to confirm deleting DefaultPartition, as shown in Figure 6-22.

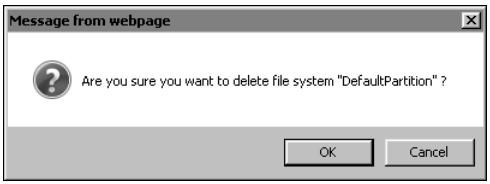


Figure 6-22 Confirming to delete DefaultPartition

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:

```
/home/actuate/Actuate/AcServer/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 6-23.

Choose OK.

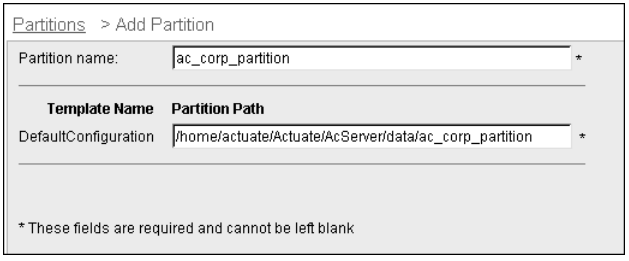


Figure 6-23 Adding a new data partition

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

7 In Template Settings, choose Test, as shown in Figure 6-24.

Choose OK.

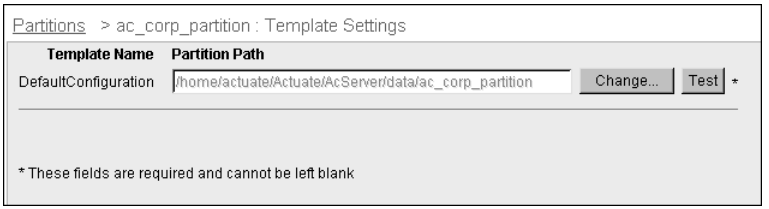


Figure 6-24 Testing a new data partition

If the test succeeds, the test successful message appears, as shown in message in Figure 6-25. Choose OK.

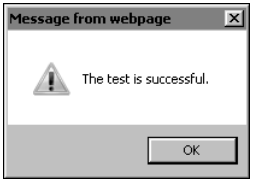


Figure 6-25 Viewing the test successful message

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 6-26.

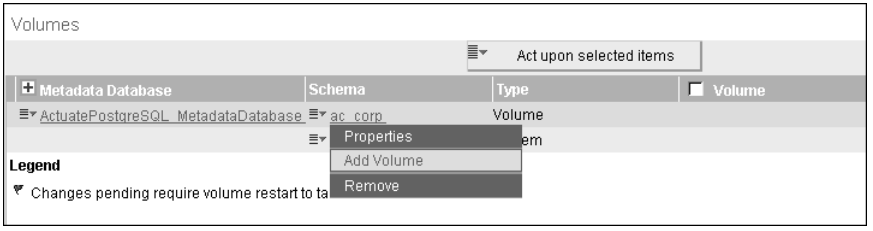


Figure 6-26 Adding a new volume

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 6-27.

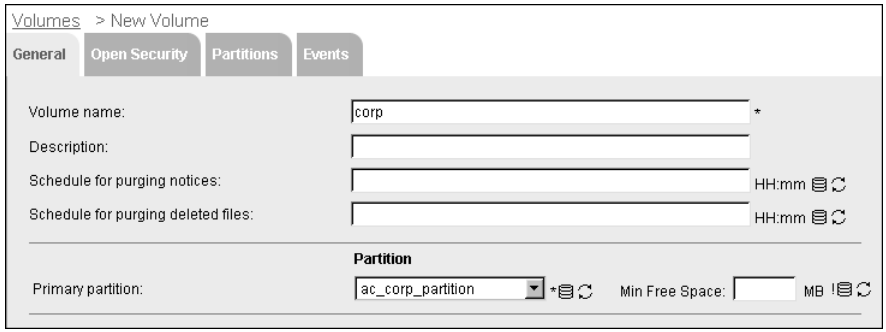


Figure 6-27 Specifying general volume properties

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 6-28.

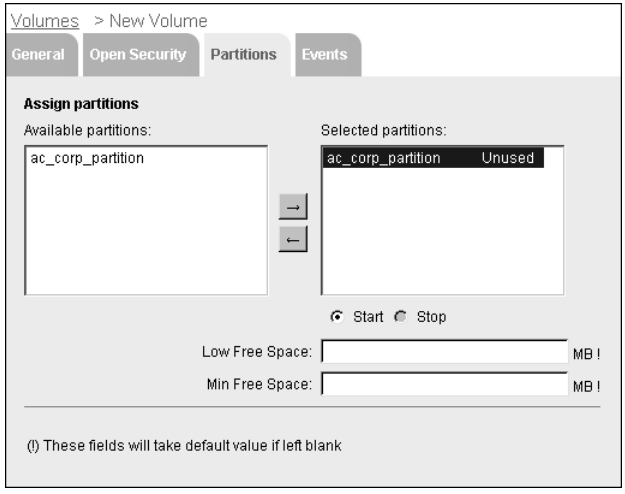


Figure 6-28 Starting the partition

Choose OK.

4 In Volumes, point to the arrow next to the new volume name, and choose Take online, as shown in Figure 6-29.

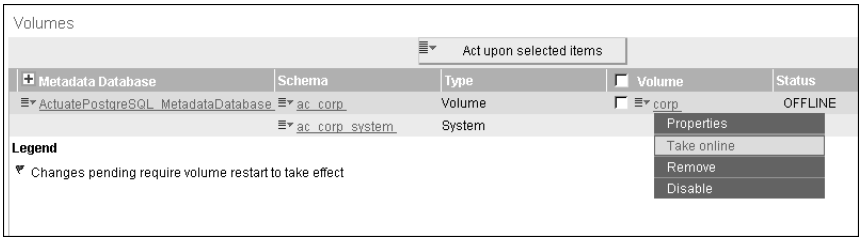


Figure 6-29 Taking a volume online

5 In Volumes, check that the status of the new volume changes to ONLINE, as shown in Figure 6-30.

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

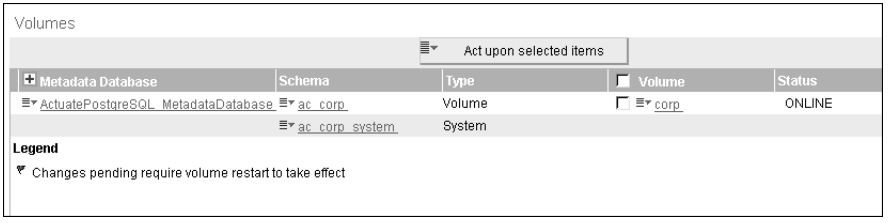


Figure 6-30 Viewing the online volume

6 Log in to Management Console to inspect the new volume by opening a browser manually and entering the following URL, as shown in Figure 6-31:

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

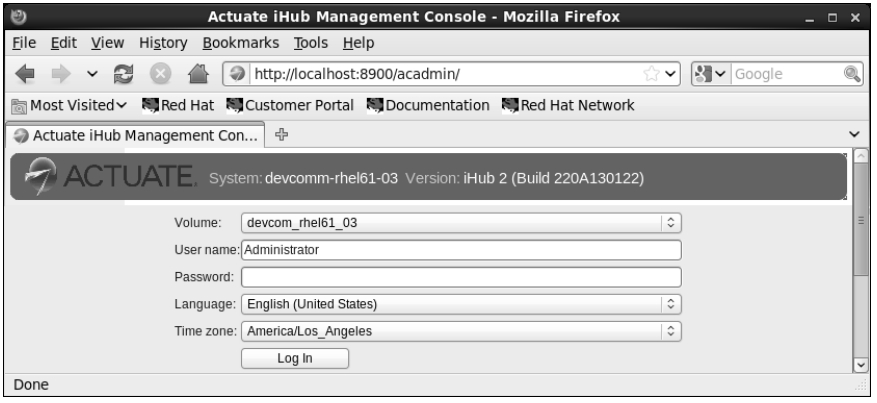


Figure 6-31 Logging in to Management Console

7 In Files and Folders, the default Encyclopedia volume appears with an empty Resources folder, as shown in Figure 6-32.

Figure 6-32 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 directory, which all nodes in the cluster access.

Before performing the cluster node installation, the administrator must also set up network sharing on the configuration folder and any required Encyclopedia volume folders, as described in "Adding a node to a cluster," in Chapter 5, "Installing a BIRT iHub cluster." It is the responsibility of the administrator performing the installation to make sure that all network sharing settings conform to the security policies in force for the environment.

This section refers to the machine containing the shared configuration directory as node1 and the cluster node accessing these shared resources 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 directory sharing and firewall settings in the network environment for the shared configuration and Encyclopedia volume folders
- 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 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.tar.gz 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, as shown in Figure 6-33. For example, type:

   `/home/actuate/Actuate/AcServer/data/config`

   Press Enter.

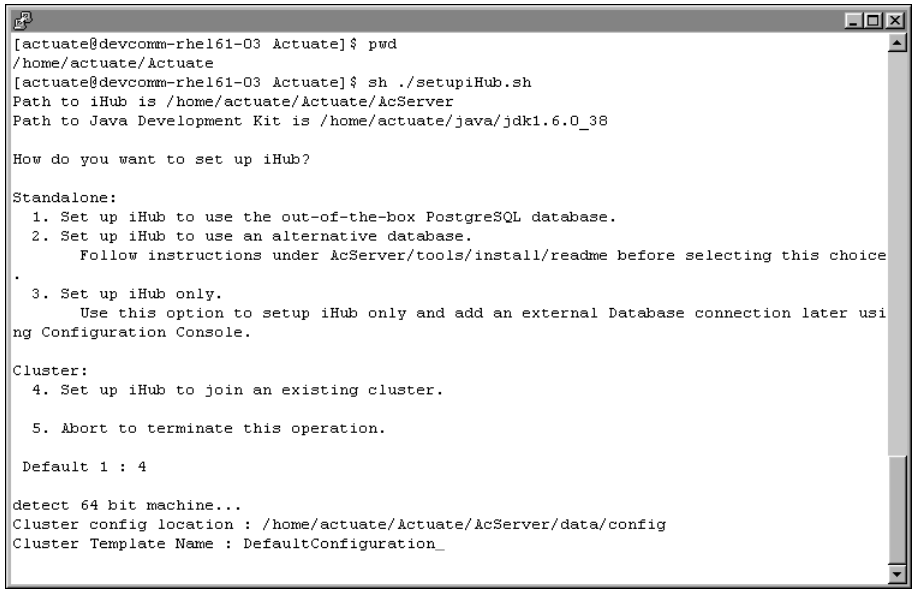


Figure 6-33 Specifying the cluster option, configuration home location, and server template name

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.

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 6-33. For example, type:

`DefaultConfiguration`

The default template name in the shared acserverconfig.xml in a cloud-based stand-alone iHub installation is DefaultConfiguration. Press Enter.

The script performs the following tasks, as shown in Figure 6-34:

- Configures iHub deployment files
- Writes setup and installation information to log files
- Generates security key files
- Sets up iHub to join the cluster
- Starts BIRT iHub 2

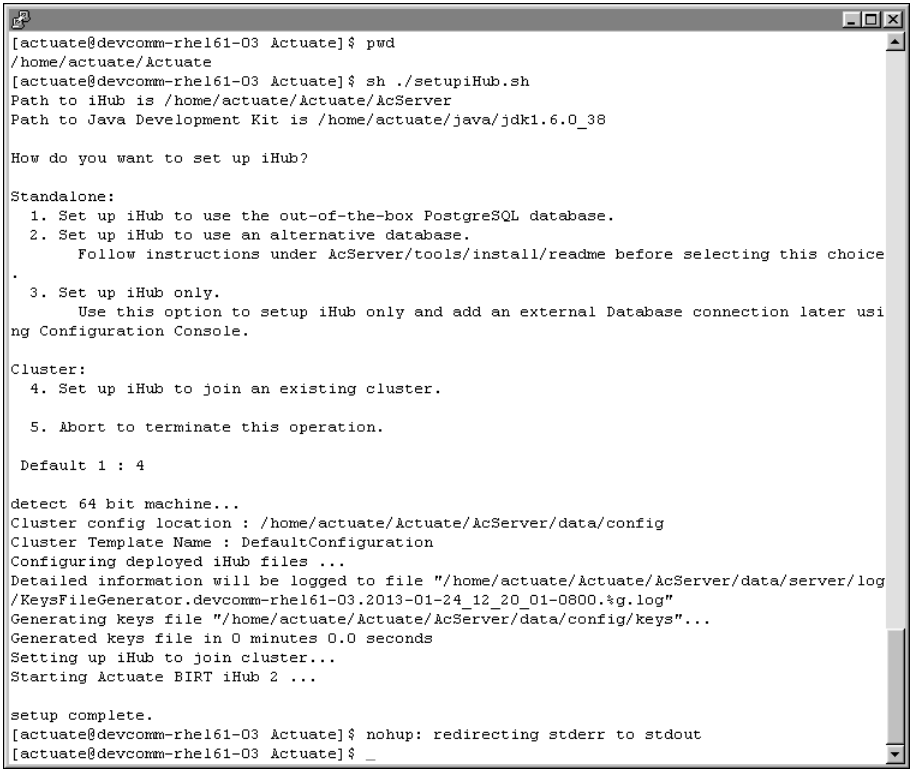


Figure 6-34 Completing the iHub cluster node setup

3 Log in to Configuration Console. In 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 6-35.

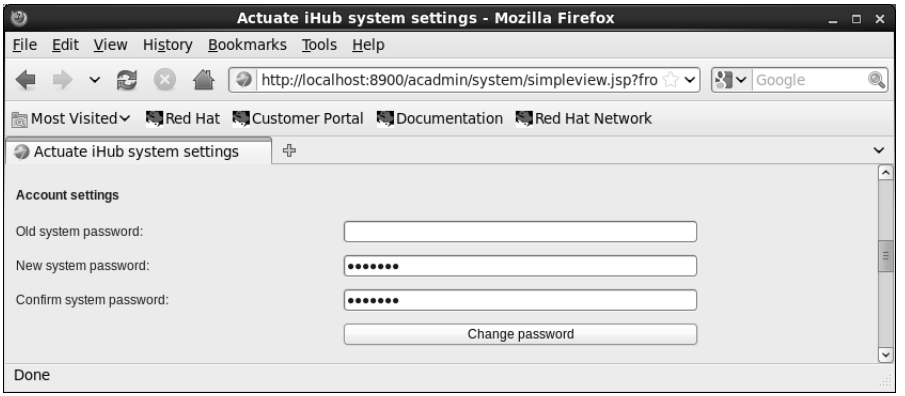


Figure 6-35 Creating a new Configuration Console password

4 Scroll to the top of Simple view and choose Advanced view. In Advanced view, choose Servers. The new node appears in the list of servers.

For more information about configuring a cluster, see Chapter 5, "Installing a BIRT iHub cluster," earlier in this book, and Chapter 9, "Clustering," in *Configuring BIRT iHub*.

Understanding the cloud deployment environment

In iHub, the location of program files changes to consolidate these resources in AC_SERVER_HOME to facilitate creating an iHub image for deployment in a cloud environment.

iServer Release 11 introduced a new environment variable, AC_DATA_HOME, for specifying the location of iServer 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 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/oda

Specifying AC_DATA_HOME

In an iServer Release 11 product installation, the following data files moved 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:

- Before you begin
- Installing Information Console on Linux

Before you begin

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 prior to an installation.

About installing from an FTP download

If you download an Actuate product from the Actuate FTP software distribution 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 beginning the installation process.

If any files are missing, the installation program exits. Files can be missing if you download the DVD image, 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

If you perform a full installation for iHub and Information Console, install BIRT iHub System products in the following order:

- BIRT iHub and Management Console
- Information Console
- BIRT iHub Integration Technology

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
```

Installing Information Console on Linux

This section describes how to install Information Console for Linux. Before you begin the installation process, ensure that you have Actuate administrator, system administrator, and web administrator privileges. If you download an Actuate product for Linux from the Actuate FTP software distribution site, keep all the files together in their original relative locations after you extract them.

Information Console installation requires the following information:

- Installation directory
  $HOME, the account's home directory, is the default installation directory.

- Port used by the Apache Tomcat Information Console service
  If you are using a firewall, ensure that the firewall passes the port number you select. The default port is 8900.
- Encyclopedia volume name that Information Console accesses
  The default is the current machine.

You can install Information Console in following ways:

- Use an installation script.
  The installation script configures Information Console, creates shortcuts, and extracts and installs all necessary files. Use this option for automated configuration.
- Deploy a WAR file to an application server.
  Deploying directly requires that you configure Information Console for your application server. Use this option if your application server supports configuration of an application from a WAR file.

Using the script to install

Complete the steps in the following section to install Information Console using the installation script.

How to install using a script

1. Download the required files from the FTP software distribution site. Extract the files.
2. To install the Information Console files, type:

   ```
   sh ./infoconsoleinstall.sh
   ```

   The script displays a series of prompts. Respond to the prompts as described in the following procedures.
3. The license agreement appears, as shown in Figure 7-1.

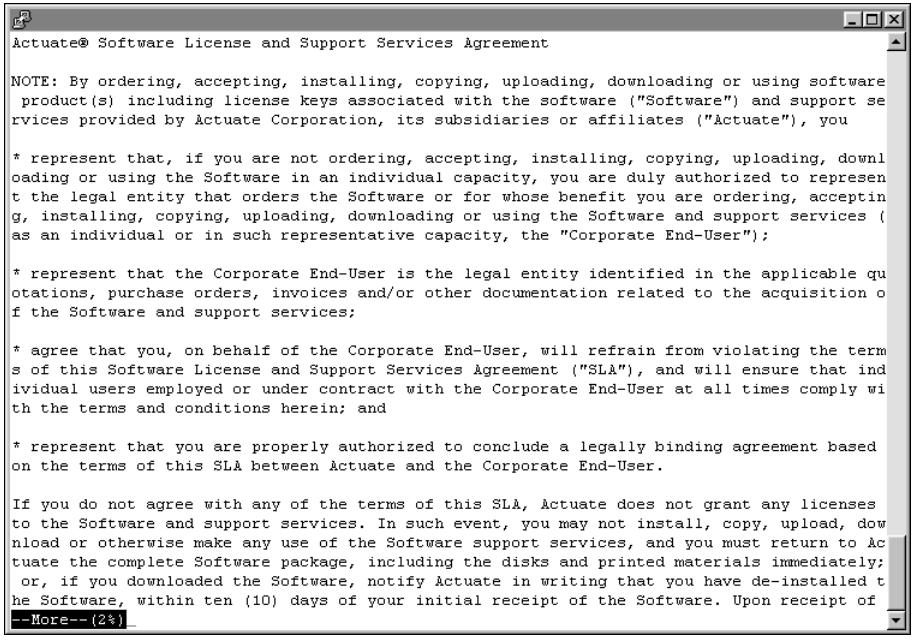


Figure 7-1 The license agreement

4 Read the license agreement and press Enter to continue installation. At the prompt, type y for yes if you accept the licensing terms, as shown in Figure 7-2.

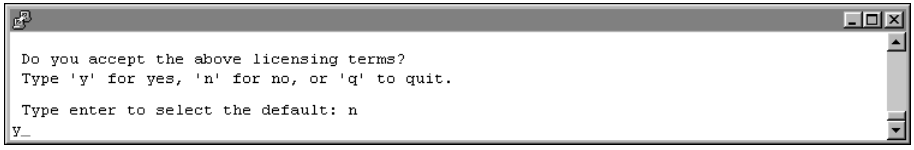


Figure 7-2 Specifying whether you accept the license agreement

5 The introduction to the installation appears, as shown in Figure 7-3. Review the information, then press Enter to continue.

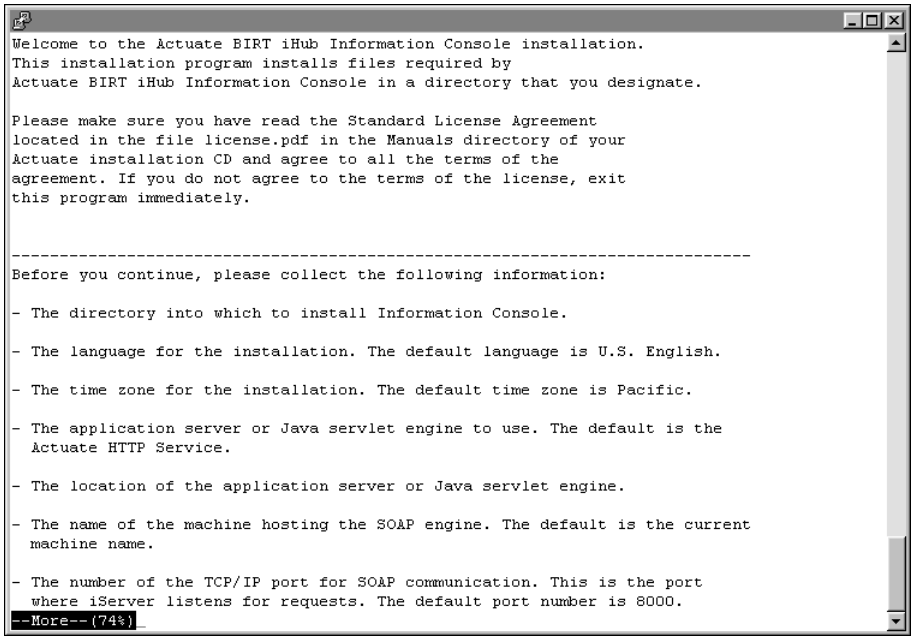


Figure 7-3 Viewing the introduction

6 Type the path for the Information Console installation, for example /home /actuate/IC, as shown in Figure 7-4. Alternatively, press Enter to accept the default directory.

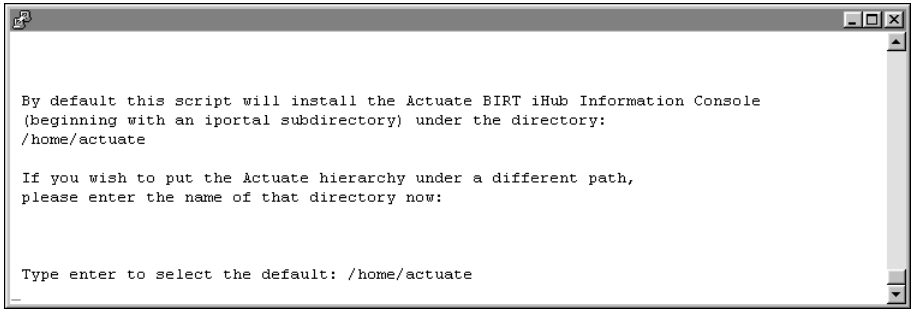


Figure 7-4 Specifying the Information Console install directory

7 Press Enter to select the default locale, which is English, as shown in Figure 7-5. Alternatively, select a different locale. If you do not see the locale for your region, type m for more and press Enter.

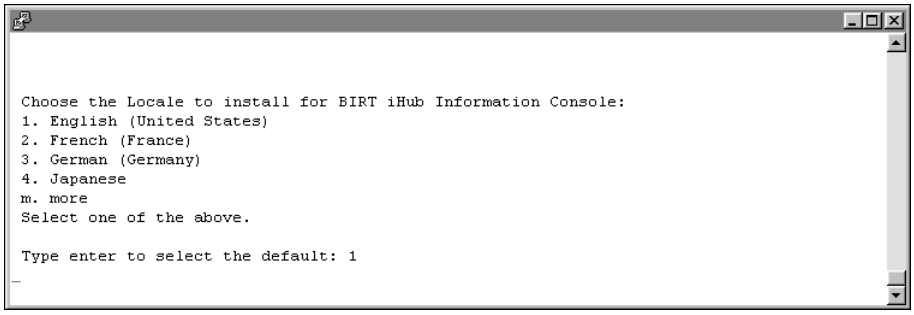


Figure 7-5 Specifying a locale

8 Press Enter to select the default time zone, which is America/Los_Angeles as shown in Figure 7-6. Alternatively, select another time zone from the numbered list.

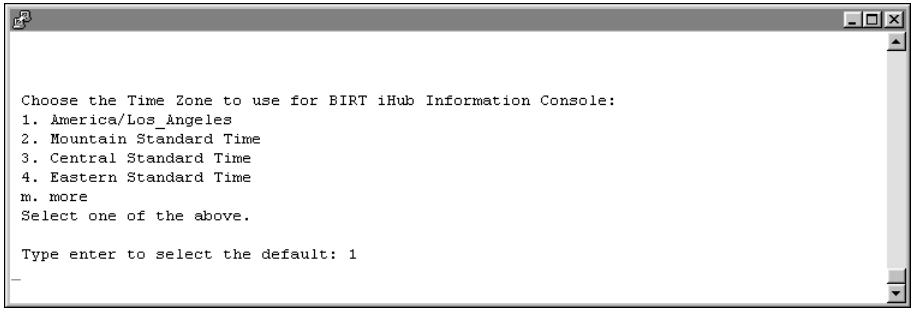


Figure 7-6 Specifying a time zone

9 Type a default profile name that you will use in Information Console, as shown in Figure 7-7.

Figure 7-7 Specifying the default profile name

10 Type the IP address or host name of the machine where iHub runs, or accept the default, your machine name, as shown in Figure 7-8.

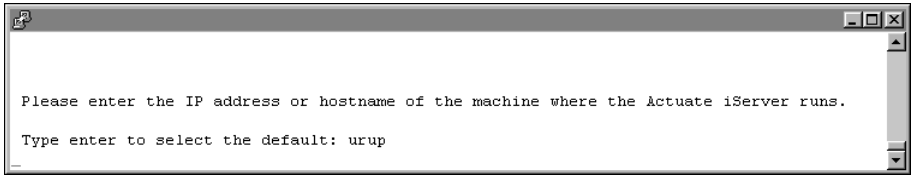


Figure 7-8 Specifying the machine on which the iHub runs

11 Type the number of the port where iHub listens for requests, or accept the default, 8000, as shown in Figure 7-9.

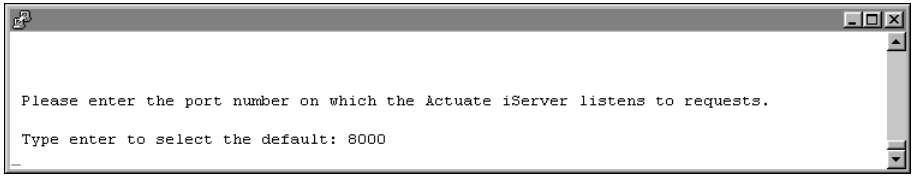


Figure 7-9 Specifying the port number on which iHub listens

12 Type the Encyclopedia volume name that you want to use, or accept the default, your machine name, as shown in Figure 7-10.

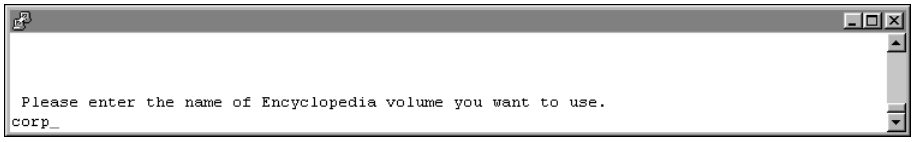


Figure 7-10 Specifying the Encyclopedia volume name

13 The installation program displays the settings that you specified during the install process. Review these settings, as shown in Figure 7-11, then specify whether you accept them. Press Enter to accept the default option, y for yes. Alternatively, type n for no, or type q to quit.

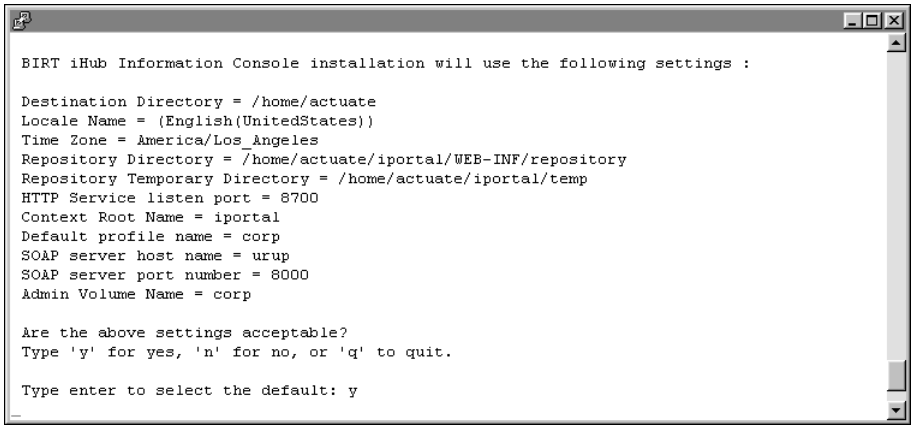


Figure 7-11 Reviewing your settings

14 The installation program installs Information Console, and displays an indicator showing the progress of the installation, as shown in Figure 7-12.

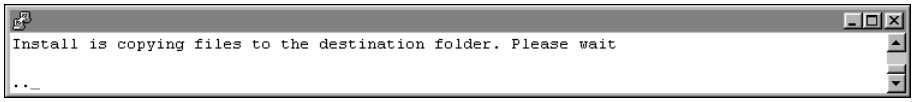


Figure 7-12 The install program copies files to your destination folder

15 When the installation completes, review the information, as shown in Figure 7-13. Issuing the command: sh ./update_rclocal_infoconsole.sh starts the Information Console service at system startup.

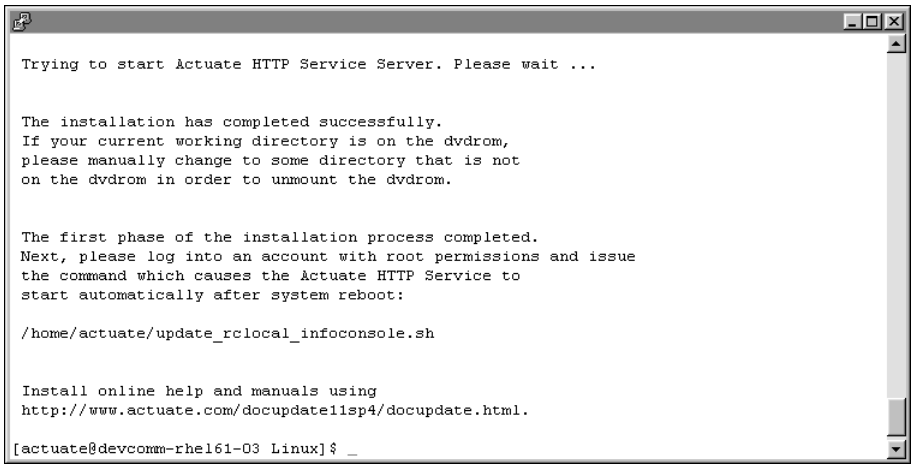


Figure 7-13 Typing the command to start the service

Using the WAR file to install

If Actuate supports your application server, you can deploy Information Console as a WAR (web archive) file. See the 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 these applications for your local environment, follow the steps in "Preparing the WAR file," later in this chapter, and 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 file and application server to integrate them. You must also verify that you have a standard J2EE installation.

To integrate the application server with Actuate, complete the following tasks:

- Configure the server for best performance 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 the application server documentation.
  - If the application server does not have deployment tools, add the Actuate context root to the application server, typically by modifying the application server configuration file.

Preparing the server

Actuate recommends the following configuration for best performance:

- Use at least a two-CPU machine for Information Console.
- If iHub 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 the locale at login.	If you change the locale, select the new locale from the locales in /WEB-INF/Localemap.xml.

(continues)

Table 7-1 Information Console configuration parameters (continued)

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 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 generic procedure for customizing an Information Console WAR file:

1. Create a temporary directory, such as /home/actuate/ic_temp.

   If you use an existing directory, ensure that this directory is empty.

2. 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:

   ```
   cp TOMCAT_ActuateInformationConsole.war /home/actuate/ic_temp
   ```

3. Decompress the WAR file, as shown in the following example:

   ```
   jar -xf TOMCAT_ActuateInformationConsole.war
   ```

   The Information Console files appear in the temporary directory.

4 Using a text editor that accepts UTF-8 encoding, edit web.xml to configure Information Console for your application server.

   If you used the temporary path in step 2, the file location is /home/actuate /ic_temp/WEB-INF/Web.xml. Refer to Table 7-1 for a list of entries to modify in web.xml.

5 Save and close web.xml.

6 Type the following command:

```
jar -cf ../newinformationconsole.war *
```

   This command creates newinformationconsole.war in the /ic_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 your application server supports clustering, see your 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 application server, if necessary.
2. Open a web browser.
3. Type the URL for the Information Console home page.

   You can use 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 URL to the BIRT iHub.

   The Information Console login page appears.

   On the Information Console login page:

   - For enterprise mode Information Console, in Volume, select an Encyclopedia Volume.
   - In User name, type your login name.
   - In Password, type your password.
   - In Language, select a locale.
   - In Time zone, select a time zone.
4. Choose Log In.

Chapter

8

Installing iHub Integration Technology and documentation

This chapter discusses the following topics:

- Installing BIRT iHub Integration Technology
- Installing the localization and documentation files
- About accessing online help

Installing BIRT iHub Integration Technology

This section describes how to install BIRT iHub Integration Technology for Linux. If you download an Actuate product for Linux from the Actuate FTP software distribution site, keep all the files together in their original relative locations after you extract them.

How to install

In a default installation, BIRT iHub Integration Technology installs in $HOME /ServerIntTech. To install BIRT iHub Integration Technology, perform the following steps:

1. Download the required files from the FTP software distribution site. Extract the files.
2. To install the server files, execute the isitinstall script:

   ```
   sh ./isitinstall.sh
   ```

   The script displays a number of prompts. Respond to the prompts as described in the following procedure.
3. The license agreement appears, as shown in Figure 8-1.

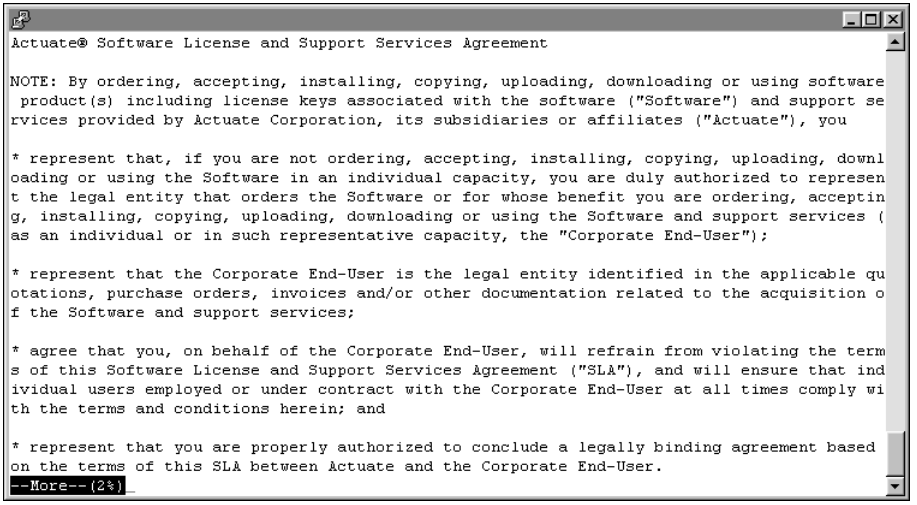


Figure 8-1 Reviewing the license agreement

4. Read the license agreement and press Enter to continue installation. At the prompt, type y for yes if you accept the licensing terms, as shown in Figure 8-2.

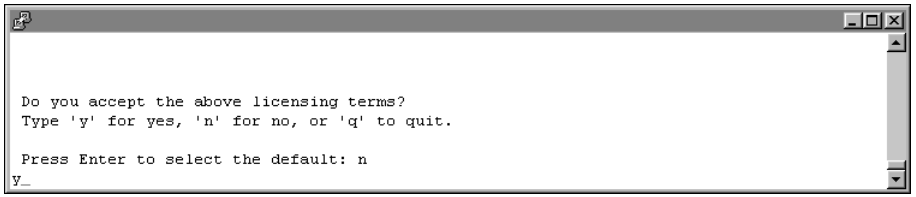


Figure 8-2 Accepting the license agreement

5 The introduction to the installation appears, as shown in Figure 8-3. Press Enter after reviewing the introductory information.

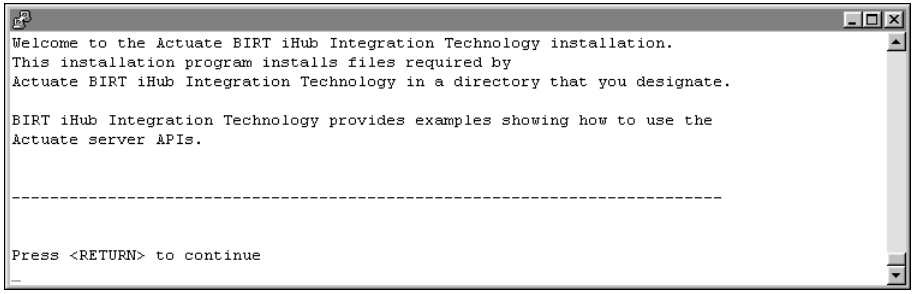


Figure 8-3 Reviewing introductory information

6 Press Enter to accept the default location for installation, $HOME /ServerIntTech as shown in Figure 8-4. Alternatively, type a different directory and press Enter.

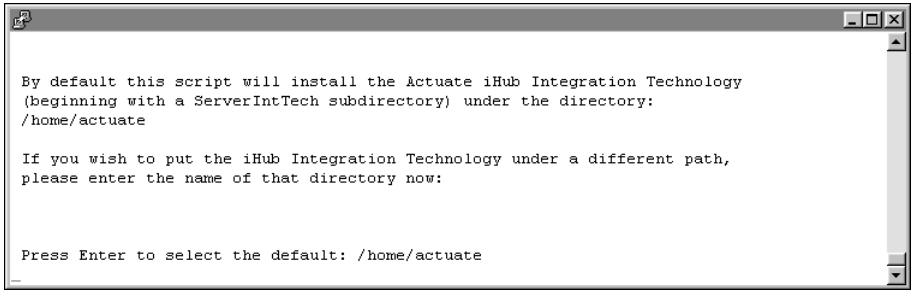


Figure 8-4 Specifying the installation directory

7 Review the settings, as shown in Figure 8-5, then specify whether to accept the settings. Press Enter to accept the default, y for yes. Alternatively type n for no, or q to quit.

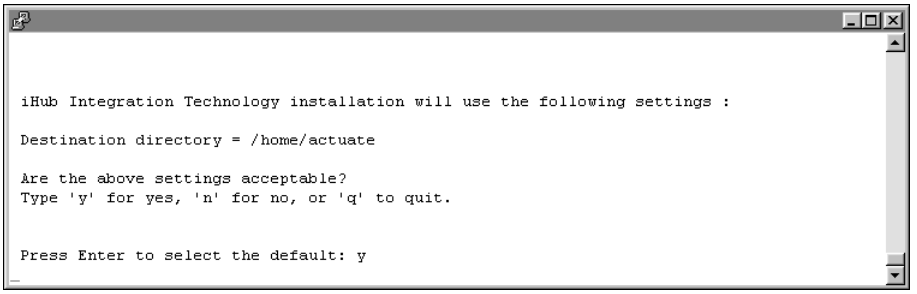


Figure 8-5 Reviewing settings before copying files

8 The installation program installs iHub Integration Technology, and displays an indicator showing how the installation is progressing, as shown in Figure 8-6.

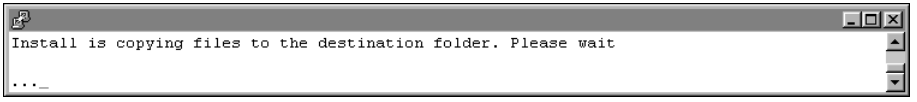


Figure 8-6 Copying files to your destination folder

9 When the installation program finishes, it provides additional information about changing the working directory and installing online help and manuals, as shown in Figure 8-7.

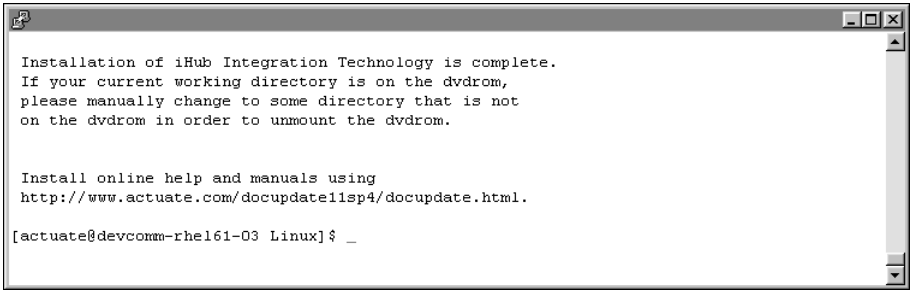


Figure 8-7 Viewing information about changing the working directory and installing online help

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 steps:

1. After downloading the tar file, decompress it using the tar command:

   ```
   tar -xvf FILE_NAME.tar
   ```

2. To begin the installation, move to the newly decompressed directory and execute the helpinstall script:

   ```
   sh ./helpinstall.sh
   ```

   The script displays a series of prompts. Respond to the prompts as described in the following procedures.

3. The license agreement appears, as shown in Figure 8-8.

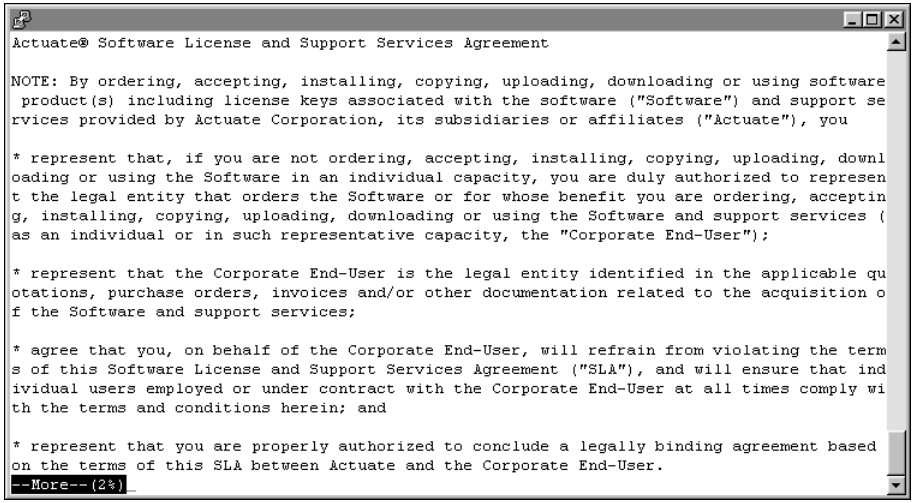


Figure 8-8 Reviewing the license agreement

4 Read the license agreement and press Enter to continue installation. At the prompt, type y for yes if you accept the licensing terms, as shown in Figure 8-9.

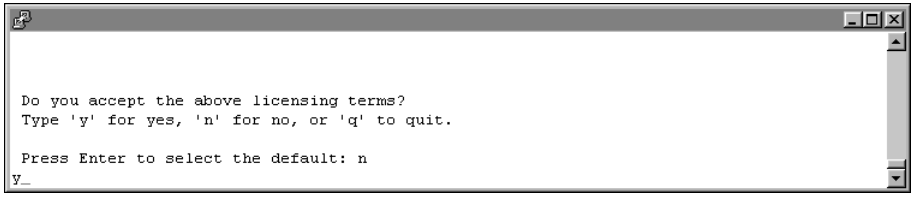


Figure 8-9 Accepting the license agreement

5 Choose the products that you wish to update in this install, as shown in Figure 8-10. If you want to choose more than one, just enter the numbers separated by a space. For example, type 1 2 3 to select all products.

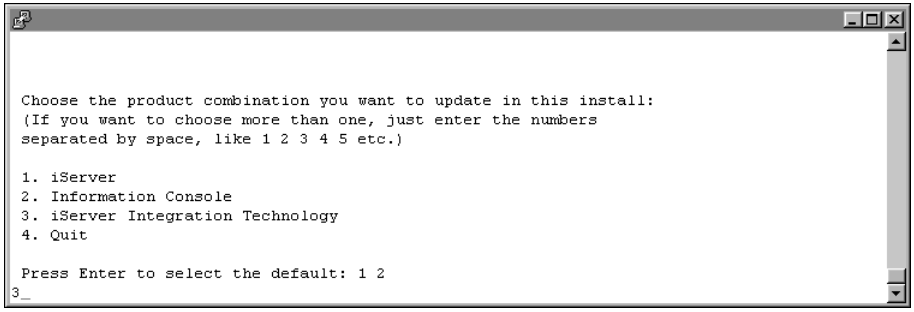


Figure 8-10 Selecting a product combination

6 Enter the full path of the product installation directory that you chose to update, as shown in Figure 8-11. If you chose to update more than one product, you are asked for the full path of every product directory in your selection.

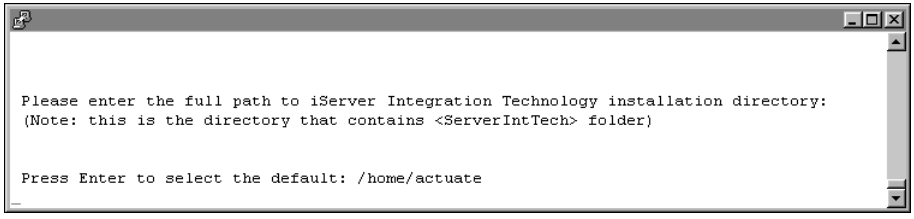


Figure 8-11 Specifying a product directory

7 Review the settings, as shown in Figure 8-12, then specify whether you accept the settings. Press Enter to accept the default, y for yes. Alternatively, type n for no, or q to quit.

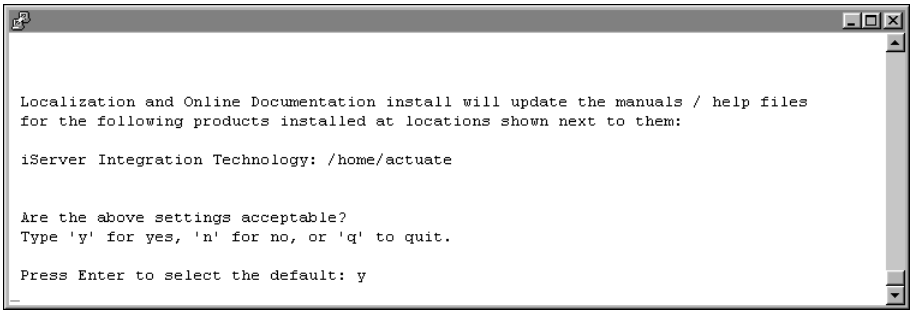


Figure 8-12 Reviewing settings before copying files

8 The installation program starts and displays an indicator showing how the installation is progressing, as shown in Figure 8-13.

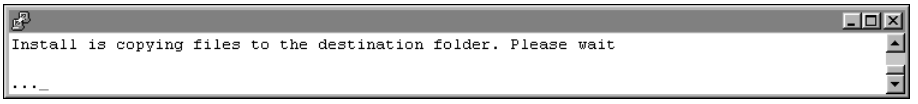


Figure 8-13 Copying files to your destination folder

9 When the installation program finishes, it provides additional information about changing the working directory, as shown in Figure 8-14.

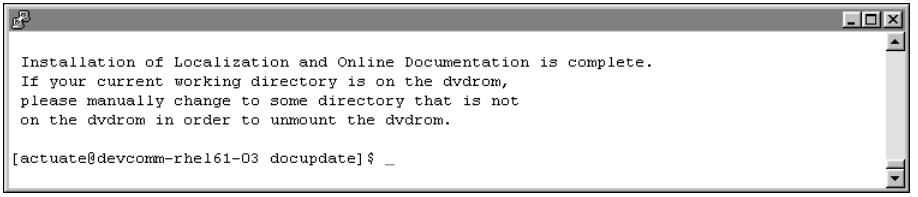


Figure 8-14 Viewing information about changing the working directory

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

After installing the iHub localization and documentation files, perform the following steps:

1. Move to the iHub installation's bin directory and execute the isdocupdate script:

   ```
   sh isdocupdate.sh
   ```

   The following text appears:

   ```
   Product help can be accessed locally or from www.actuate.com
   Please select your preference:
   1. Use online help
   Recommended: Access the help from www.actuate.com. It ensures
      you always have the latest information
   2. Use local help
   Use a locally installed copy of the help content (may require a
      separate installation of the help content). Only needed if
      you do not have an internet connection.
   Select one of the above.

   Press Enter to select the default: 1
   ```

2. To use locally installed help files, type 2. To revert to using online help files from the Actuate web site, type 1.

3. Restart BIRT iHub System. Restarting applies the changes to the online documentation location.

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 an BIRT iHub System. A named user is a distinct individual who receives content and value from BIRT iHub.

  A BIRT 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 onDemand 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 convenient 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

(continues)

Table 9-1 BIRT iHub System license options (continued)

Option	Description	Supported releases
BIRT Studio	Allows a user to create a BIRT design and to run it in BIRT iHub. BIRT Studio Option supports access to an information object on BIRT iHub System.	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. Access privileges are based on user name or security role.	11, iHub
Multi-Tenant	Allows a BIRT iHub System user to access more than one Encyclopedia volume. This option is available with an Unlimited User CPU License.	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.

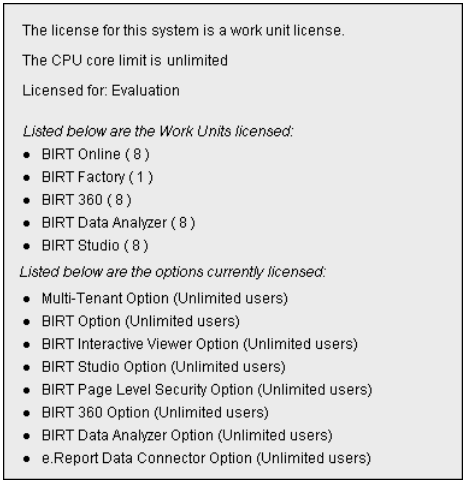


Figure 9-1 iHub License options

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 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 BIRT iHub System.

After installing Actuate 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 install program prompts you to provide the location of the license file. After providing the location of the license file, the install program 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\AcServer\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.

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. Navigate to AC_SERVER_HOME\bin.
2. Type the following command and press Enter:

   ```
   sh ./acmachineid.sh
   ```

   The utility provides output in the following format:

   ```
   STATUS:          OK
   GEN_VERSION:     22 iHub
   GEN_BUILD:       XxXBuild NumberXxX
   MACHINEID:
                    I0RREHs0Jk6tu0o8AbCrVL61x7kDpLlQKwS2t1W7qM67GbO8
                    VjcFs6pcuAgbtDaZauSbFFa2mRejwVJc7ZjKfMEVl1suXglM
                    KmZLiwtLykwJisqMS0EhYe5sCYoKjG+XL2UEnL2GGhLtI9f
                    JUMYzZORKk23jrxaSwUDsgKsvlc1A6q8UbmrrAYHD8Ggtpui
                    AmxWt4xjEM6rqlmsNEW/4ViMC0KDBkSn
   ```

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 a .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.

The 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 an 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.

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.

Configuring CPU binding on Linux

The following section describes how to perform various CPU-binding operations in the Linux environment.

The pmdXX process, where XX refers to the program version number, is the root parent process for all other iHub processes, so CPU binding can be done only for pmdXX. Binding must be done before starting the pmdXX process. Binding a running pmdXX process will not take effect.

How to configure CPU binding on Linux

1 In Linux, log in as root and use the less command to view CPU and core information, as shown in the following example:

```
less /proc/cpuinfo
```

Use <Ctrl>+<Z> suspend the command.

2 To verify CPUs/cores, use the cat command for more detailed information, as shown in the following example:

```
sudo cat /proc/cpuinfo.
```

3 Use the taskset command, referencing the process ID (PID) to verify processor affinity, as shown in the following example:

```
taskset -p -c PID
```

Using the -c option provides a processor affinity mask in list form rather than a bit mask for a PID specified using the -p option. For example, a typical affinity list generated by these arguments is 0,2,3-5.

4 Use the ps|grep commands to verify the current core settings for all running processes, as shown in the following example:

```
ps -e -o pid,cpuid,comm,user | grep clnqa
```

5 iServer (pmd) <Stop> automatically removes a previous processor binding.

6 To bind CPUs on Linux perform the following tasks:

   1 Stop iHub, including the PMD. Make sure no processes are running by typing the following command:

   ```
   ps -e -o pid,cpuid,comm,user | grep clnqa
   ```

2 In a typical installation process, using a text editor such as vi, open the pmd11.sh file located in AcServer/bin and edit the following strings.

Change:

```
$g_AcRSMyFullPath/$daemon > /dev/null &
```

to:

```
taskset 3 $g_AcRSMyFullPath/$daemon > /dev/null &
```

where taskset 3 is a bit mask representing core #0 and core #1.

Alternatively, when using an archive installation process, using a text editor such as vi, open the startsrvr.sh file located in AcServer/bin and edit the following strings.

Change:

```
nohup "$AC_SERVER_HOME/bin/pmd11" > $AC_DATA_HOME/server/log
   /startsrvr.out  &
```

to:

```
taskset 3 nohup "$AC_SERVER_HOME/bin/pmd11" > $AC_DATA_HOME
   /server/log/startsrvr.out  &
```

3 Restart iHub. Make sure that all processes started and verify the core to which each process is bound by running the following command:

```
ps -e -o pid,cpuid,comm,user | grep clnqa
```

The <taskset> command binds to the number of logical cores. If a machine does not have hyperthreading enabled, you will not see any difference between the physical and logical cores.

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 volume 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," later in this book.

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 Linux BIRT iHub environment, the default AC_SERVER_HOME path is:

```
/home/actuate/AcServer/
```

The default AC_DATA_HOME path is:

```
/home/actuate/AcServer/data/
```

The default Encyclopedia volume path is:

```
/home/actuate/AcServer/data/encyc
```

The default acserverconfig.xml file path is:

```
/home/actuate/AcServer/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, 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

   Choose OK.

For information on other aspects of archiving, see Chapter 12 "Archiving files," in *Configuring BIRT iHub*.

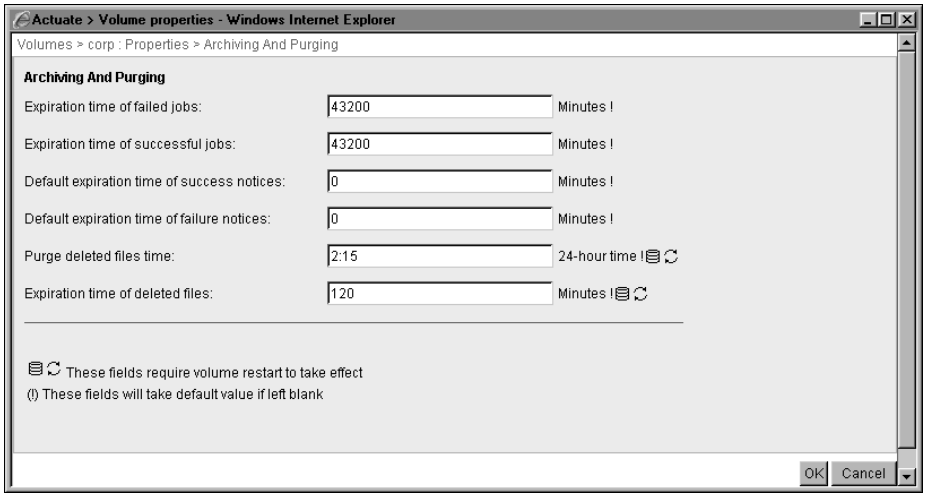


Figure 10-1 Configuring file purging properties

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 pg_dump

To back up an Encyclopedia volume using the pg_dump 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 pg_dump utility.
- Back up the acserverconfig.xml file and volume data folders to the backup folder.

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.

In a Linux 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. Navigate to your home folder, which by default is:

   ```
   /home/actuate
   ```

2. Create the following new folder, as shown in Figure 10-2:

   ```
   /home/actuate/encyc_backup
   ```

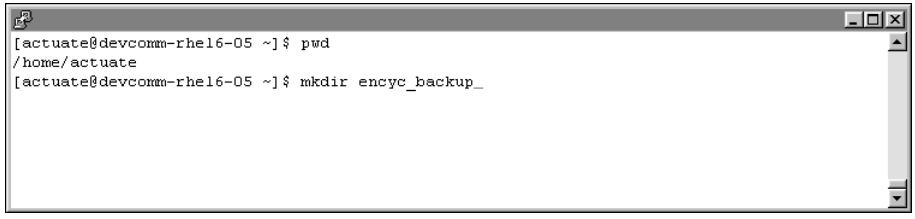


Figure 10-2 Creating a backup folder

Back up Encyclopedia volume metadata using pg_dump by performing the following tasks.

How to run pg_dump

1. Navigate to the following location:

   ```
   /home/actuate/AcServer/postgresql/bin
   ```

2 Execute the following command. Substitute your machine name for devcomm-lnx-rh-01 in this example:

```
./pg_dump --host devcomm-rhel6-05 --port 8432 --username
   postgres --format custom --blobs --verbose --file "/home
   /actuate/AcServer/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.

3 The command line appears as shown in Figure 10-3.

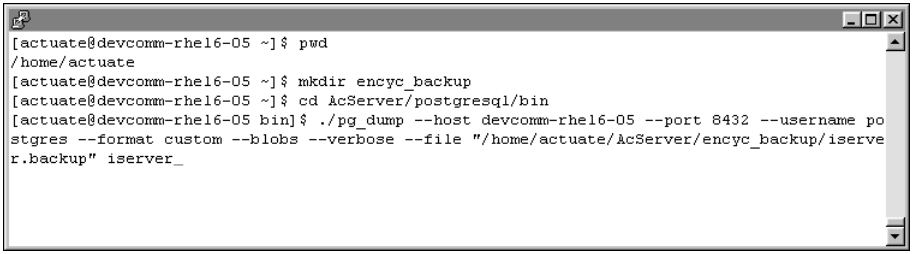


Figure 10-3 Entering the command to execute pg_dump

4 Type the postgres superuser password. The administrator specified this password 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 Navigate to AC_DATA_HOME, which is the location of the iHub data. You specified this location on Setup Type during the install procedure. The default path for AC_DATA_HOME is:

```
/home/actuate/AcServer/data
```

2 In AC_DATA_HOME, navigate to the config folder that contains acserverconfig.xml file.

Copy acserverconfig.xml to the following backup location, as shown in Figure 10-4:

```
/home/actuate/encyc_backup
```

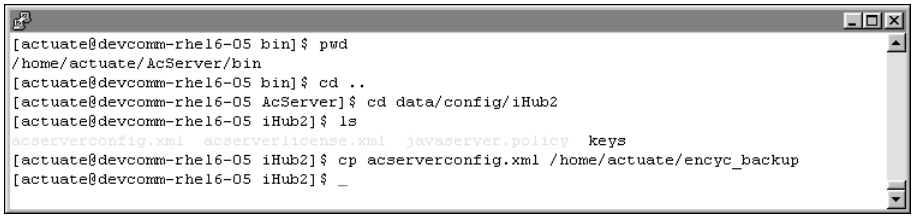


Figure 10-4 Copying acserverconfig.xml to the backup location

3 Navigate to AC_SERVER-HOME/encyc.

Copy the file and filetype folders, and status folder, if it exists, to the following backup location, as shown in Figure 10-5:

```
/home/actuate/encyc_backup
```

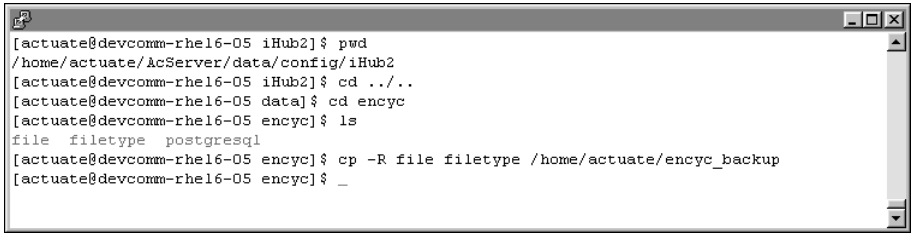


Figure 10-5 Copying the volume data folders to the backup location

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 the 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.

The contents of the backup folder appears, as shown in Figure 10-6.

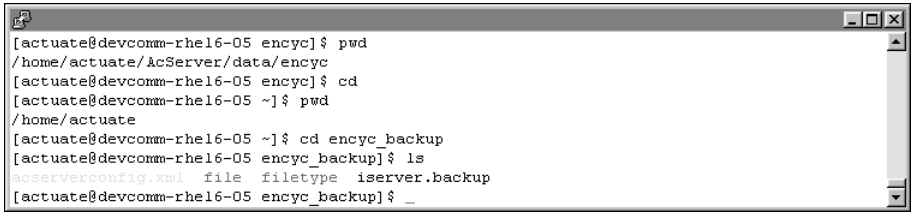


Figure 10-6 Viewing the contents of the backup folder

Restoring an Encyclopedia volume using pg_restore

To restore a backed up Encyclopedia volume, perform the following tasks:

- Take the Encyclopedia volume offline.
- Delete the acserverconfig.xml 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.

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 by performing the following tasks.

How to take the Encyclopedia volume offline

1 In a web browser type:

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

Log in to Configuration Console as Administrator.

2 In the simple view, choose Advanced view. Choose Volumes.

3 On Volumes, take the volume offline, as shown in Figure 10-7.

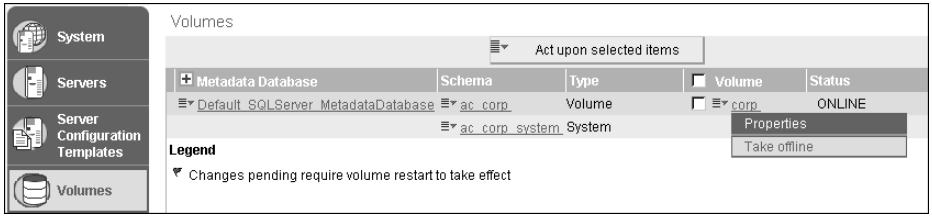


Figure 10-7 Taking the volume offline

How to restore the backed up volume data folders

1 Navigate to the AC_DATA_HOME/config/iHub2 folder that contains the acserverconfig.xml file.

Delete acserverconfig.xml, as shown in Figure 10-8.

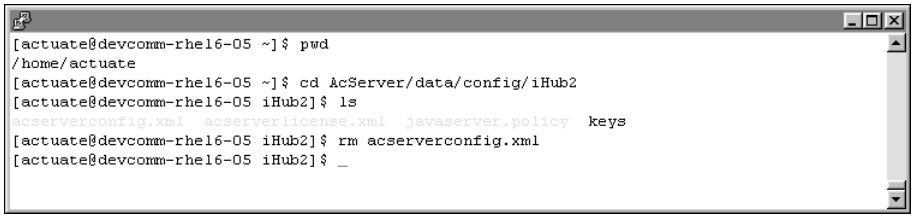


Figure 10-8 Deleting acserverconfig.xml

2 In AC_DATA_HOME, open the encyc folder.

In AC_DATA_HOME/encyc, delete the file and filetype folders, and status folder, if it exists, as shown in Figure 10-9.

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 delete the postgresql folder.

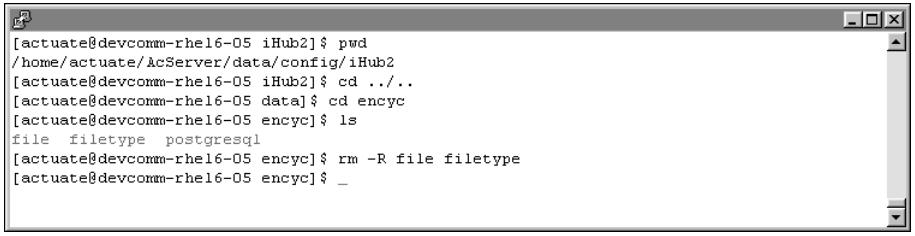


Figure 10-9 Deleting the file and filetype folders

3 Navigate to the following location:

`/home/actuate/encyc_backup`

Copy acserverconfig.xml to the appropriate AC_DATA_HOME/config /iHub2 folder, as shown in Figure 10-10.

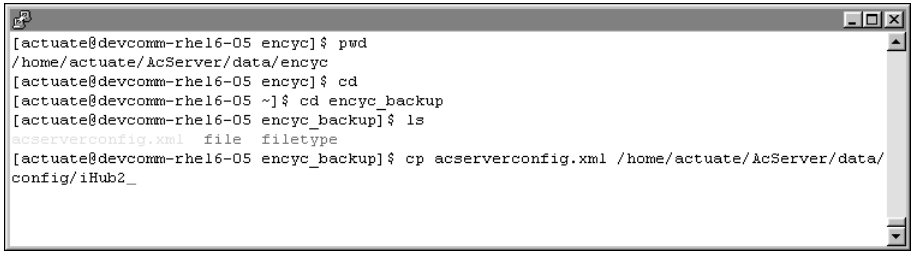


Figure 10-10 Copying acserverconfig.xml to AC_DATA_HOME/config/iHub2

The contents of AC_DATA_HOME/config/iHub2 appear as shown in Figure 10-11.

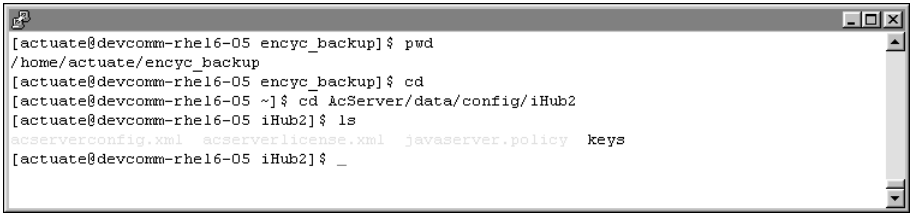


Figure 10-11 Viewing the contents of AC_DATA_HOME/config/iHub2

4 Copy the file and filetype folders, and status folder, if it exists, to AC_DATA_HOME/encyc, as shown in Figure 10-12.

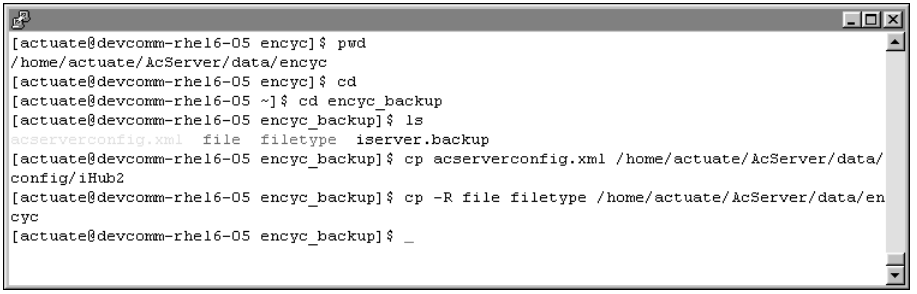


Figure 10-12 Copying the volume data folders to AC_DATA_HOME/encyc

The contents of AC_DATA_HOME/encyc appear as shown in Figure 10-13.

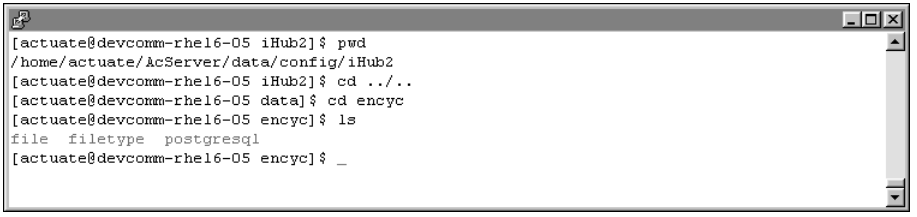


Figure 10-13 Viewing the contents of AC_DATA_HOME/encyc

How to run pg_restore

1 Navigate to the following location:

```
/home/actuate/AcServer/postgresql/bin
```

2 Execute the following command. Substitute your machine name for devcomm-lnx-rh-01 in this example:

```
./pg_restore --host devcomm-rhel6-05 --port 8432 --username
  postgres --dbname iserver --clean --verbose "/home/actuate
  /AcServer/encyc_backup/iserver.backup"
```

3 The command line appears as shown in Figure 10-14.

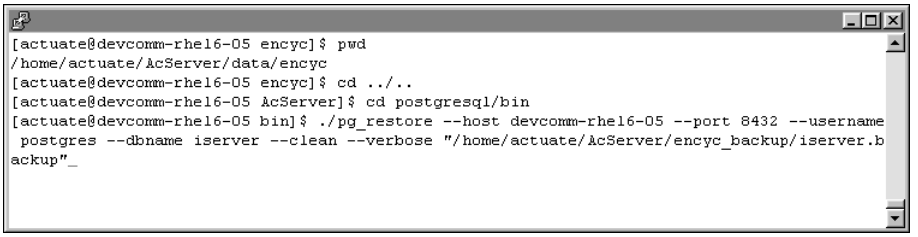


Figure 10-14 Entering the command to execute pg_restore

How to take the Encyclopedia volume online

1. In a web browser, type:

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

   Log in to Configuration Console as Administrator.

2. In the simple view, choose Advanced view. Choose Volumes.

3. On Volumes, take the volume online, as shown in Figure 10-15.

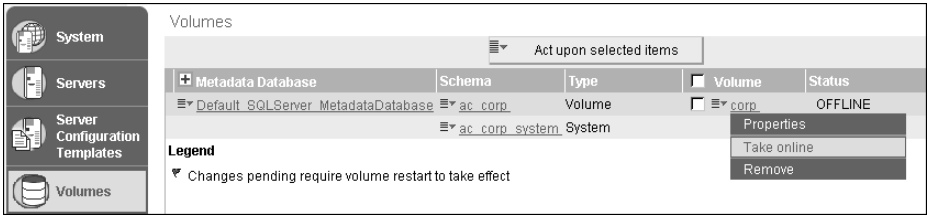


Figure 10-15 Taking the volume online

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 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

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 whitespace automatically, but preserve embedded whitespace.

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 migration 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.
DATABASE_TYPE	Type of supported RDBMS that contains the data store. Specify DB2, Oracle, or PostgreSQL.

Table 11-1 Required Encyclopedia Data Store Administrator properties

Parameter	Description
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 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 'Oracle' or 'PostgreSQL'.	{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, 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_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 relational database system that contains the data store. iHub currently supports DB2, Oracle, and PostgreSQL in Linux.		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
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

(continues)

Table 11-2 Optional Encyclopedia Data Store Administrator properties (continued)

Parameter	Description	Default value	Supported databases
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 is appended 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
LOG_FOLDER	Full path of folder to write logs.	AC_DATA _HOME /server/log.	

Table 11-2 Optional Encyclopedia Data Store Administrator properties (continued)

Parameter	Description	Default value	Supported databases
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
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 structures in database.	POPULATE _SCHEMA	All

(continues)

Table 11-2 Optional Encyclopedia Data Store Administrator properties (continued)

Parameter	Description	Default value	Supported databases
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.	NEW _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
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 import and export 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 = /home/actuate/iHub/AcServer
DATABASE_TYPE = PostgreSQL
DEFAULT_DATABASE_NAME = postgresSUPERUSER = postgres
SUPERUSER = postgres
SUPERUSER_PASSWORD = <your superuser 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 = {ISERVER OR IHUB_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 = /home/actuate/iHub/AcServer
DATABASE_TYPE = PostgreSQL
SUPERUSER = postgres
SUPERUSER_PASSWORD = <your superuser 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 = /home/actuate/iHub/AcServer
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 = home/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 = /home/actuate/iHub/AcServer
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 = home/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 = /home/actuate/iHub/AcServer
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 = /home/actuate/iHub/AcServer
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 = /home/actuate/iHub/AcServer
```

```
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 = /home/actuate/iHub/AcServer
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 = /home/actuate/iHub/AcServer
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 a new volume in a new schema

This operation is only supported for PostgreSQL. iHub typically performs this operation when you create a new 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 = /home/actuate/iHub/AcServer
DATABASE_TYPE = PostgreSQL
DATABASE_NAME = iserver
DATABASE_HOST = localhost
DATABASE_PORT = 8432
DEFAULT_DATABASE_NAME = postgres
SUPERUSER = postgres
SUPERUSER_PASSWORD = <your superuser 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
```

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 = /home/actuate/iHub/AcServer
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 metadata. The system schema is a required element for any iHub installation. In a cluster, the nodes share the system schema metadata and use this information to communicate and coordinate processing.

In iHub, it is not necessary to back up the iHub 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.

Use the System Data Store Administrator utility to perform the following tasks:

- Create or delete a schema.
- Populate or depopulate a schema.
- Import or export data.

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:

```
/home/actuate/iHub/AcServer/bin
```

To run the System Data Store Administrator utility, perform the following tasks:

1. 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 chapter. You pass this file to the System Data Store Administrator utility when you execute it.
3. Create a script or run the administrate_system_data_store.sh script, passing in the name of the properties files as an argument, using the following command line syntax:

   ```
   sh ./administrate_system_data_store.sh systemdatastore
      .properties
   ```

   Alternatively, you can execute the utility by running the upgrdsds.sh script, using the following command-line syntax:

   ```
   sh ./adminsds systemdatastore.properties
   ```

   The script 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.sh 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.sh

```
#!/bin/sh

if [ "x$1" = "x" ]; then
   echo "Usage: administrate_system_data_store.sh <properties
   file name>"
   exit 1
fi

# Set up environment variables
. `dirname $0`/set_tools_environment.sh

# Administrate data store
java com.actuate.iserver.system.datastore.admin
   .SystemDataStoreAdministrator "$PROPERTY_FILE"
```

The SystemDataStoreAdministrator class has the same parent class as the Encyclopedia Data Store Administrator and uses the same property settings. 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, and superuser names 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

This operation 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 = /home/actuate/iHub/AcServer
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 Encyclopedia Data Store Upgrader

Use Encyclopedia Data Store Upgrader to upgrade 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.
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_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

(continues)

Table 11-4 Optional Encyclopedia Data Store Upgrader 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.	False	All
DATABASE _INSTANCE	RDBMS instance that manages the database.		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
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

Table 11-4 Optional Encyclopedia Data Store Upgrader properties (continued)

Parameter	Description	Default value	Supported databases
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
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

Numerics

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