Sterling Commerce
An IBM Company

Installation Guide

Release 7.5 SP1

May 2006

Copyright Notice

Installation Guide, Release 7.5 SP1

Copyright © 2000 - 2006
Yantra Corporation
ALL RIGHTS RESERVED

WARNING: ANY UNAUTHORIZED DUPLICATION OF THIS DOCUMENTATION SHALL BE AN INFRINGEMENT OF COPYRIGHT

Trade Secret Notice

This documentation, the software it describes, and the information and know-how they contain constitute the unpublished, proprietary, confidential and valuable trade secret information of Yantra Corporation, its affiliated companies or its or their licensors, and may not be used for any unauthorized purpose, or disclosed to others without the prior written permission of the applicable Yantra entity. This documentation and the software it describes have been provided pursuant to a license agreement that contains prohibitions against and/or restrictions on their copying, modification and use. Duplication, in whole or in part, if and when permitted, shall bear this notice and the Yantra Corporation copyright notice.

This documentation and the software it describes are licensed either "AS IS" or with a limited warranty, as set forth in the Yantra license agreement. Other than any limited warranties provided, NO OTHER WARRANTY IS EXPRESSED AND NONE SHALL BE IMPLIED, INCLUDING THE WARRANTIES OF MERCHANTABILITY AND FITNESS FOR USE OR FOR A PARTICULAR PURPOSE. The applicable Yantra entity reserves the right to revise this documentation from time to time and to make changes in content hereof without the obligation to notify any person or entity of such revisions or changes.

Yantra does not warrant or represent that use of the software described in this documentation will insure compliance with the U.S. Department of Transportation's (DOT) Hazardous Materials Regulations (HMR) found in Title 49 of the Code of Federal Regulations (49 CFR), and users of this software should consult independent legal counsel and technical support to insure compliance with the HMR and other legal requirements.

U.S. Government Restricted Rights. This documentation and the software it describes are each and collectively a "commercial item" as defined in 48 C.F.R. 2.101. Consequently, as and when provided to any agency or instrumentality of the U.S. Government or to a U.S. Government prime contractor or a subcontractor at any tier ("Government Licensee"), the terms and conditions of the customary Yantra commercial license agreement are imposed on Government Licensees per 48 C.F.R. 12.212 or § 227.7201, as applicable, or through 48 C.F.R. § 52.244-6.

Yantra and the Yantra logo are marks of Yantra Corporation. All other services and products or company names are used for identification purposes only and may be marks of their respective owners.

Yantra Corporation
One Park West
Tewksbury, MA 01876
1-978-513-6000

Copyright © 2000 - 2006

Contents

Preface

Intended Audience xi
Structure xi
Yantra 7x Documentation xiii
Conventions xiv

1 Getting Started

1.1 Before You Begin 1
1.2 Installation Checklist 1

2 System Requirements

2.1 Technical Stack Matrix 4
2.2 Database Tier Requirements 5
2.3 Application Server Requirements 6
2.4 Internet Browser and Plugin Requirements 7
2.5 Third-Party Software Requirements 7
2.6 Yantra 7x Mobile Application Requirements 8

3 Creating a Security Plan

3.1 Planning Your Deployment Architecture 12
3.1.1 Current Security Infrastructure Analysis 13
3.1.2 Authentication and Authorization 13
3.1.3 Data Encryption 14
3.1.4 Network Topology 14

3.1.4.1 Deployment Over the Internet 14
3.1.4.2 Deployment Over a Virtual Private Network 15
3.1.4.3 Deployment Over a Local Area Network 15
3.2 Java Protocol Security Measures 15
3.2.1 Disabling Java Protocols 16
3.2.2 Securing Java Protocols 16
3.3 Web Security 18
3.3.1 Post Installation Recommendations 18
3.3.2 Session Security 19
3.3.3 Operating System Permissions 19
3.3.4 Documentation 19
3.3.5 Routing 19
3.3.6 Web Server Executables 20
3.4 Database Security 20
3.4.1 Credit Card Encryption 20
3.5 Internet Explorer Security Settings 21
3.5.1 Configuring Browser Security Settings 21
3.5.2 Adding Yantra 7x as a Trusted Website 23

4 Installing Yantra 7x

4.1 Before You Begin 26
4.2 Installing Yantra 7x on UNIX and Linux 26
4.3 Installing Yantra 7x on Windows 27
4.4 Installing Yantra 7x on a Remote Computer 28
4.5 Uninstalling Yantra 7x 29

5 Installing the Yantra 7x Language Pack

5.1 Installing the Language Pack 32
5.2 Setting up Properties 33
5.3 Loading the Yantra 7x Language Pack Factory Defaults 33
5.3.1 Loading the Yantra 7x Language Pack Translations 34
5.3.2 Switching the Yantra 7x Base Language 34
5.4 Creating and Deploying the Enterprise Archive 35

6 Installing and Configuring Application Tier Software

6.1 Installing Your Application Server 38
6.2 Installing and Configuring Application Server Utilities 38
6.2.1 Installing JDK Upgrades 39
6.3 Installing and Configuring your Proxy Server 39

7 Installing and Configuring Database Tier Software

7.1 Installing Oracle 42
7.1.1 Oracle Database User Privileges 43
7.1.2 Configuring an Oracle database for Production 44
7.1.2.1 Running Scripts for an Oracle database 45
7.2 Installing DB2 46
7.2.1 DB2 Database User Privileges 47
7.2.2 Configuring a DB2 Database for Production 47
7.2.2.1 Running Scripts for a DB2 Database 48
7.3 Installing Microsoft SQL Server 48
7.3.1 SQL Server Database User Privileges 49
7.3.2 Configuring a SQL Server Database for a Production Environment 49
7.3.2.1 Running Scripts for a SQL Server database 49
7.4 Database Sizing 50
7.4.1 Capacity Planning 50
7.4.2 Disk Estimation for the Distributed Order Management Module 51
7.4.3 Disk Estimation for the Networked Warehouse Management System Module 54
7.4.4 Tracking and Estimating Future Disk Requirements 59

8 Installing the Print Server

8.1 Installation of Loftware Components 62
8.2 Define Printers on Loftware 63
8.3 Define Printers for Yantra 7x WMS Installation 63
8.4 Copying Yantra 7x WMS Standard Label Formats 63
8.5 Installation of JasperReports 63

9 Installing the Weighing Scale

9.1 Installation of the Weighing Scale 66

10 Installing the Yantra 7x Mobile Application

10.1 Installation on Mobile Terminals 68
10.1.1 Ensuring Re-installation on Cold Boot 71
10.2 Installing on VT220 Mobile Terminals 73
10.2.1 Installing ncurses on HP-UX 73
10.2.2 Installing libiconv on HP-UX Itanium 73
10.2.3 Installing the Yantra 7x Mobile Application on VT220 Mobile Terminals. 74

11 Configuring Yantra 7x Properties

11.1 Setting Up the Properties Files 80
11.2 Properties for the Yantra 7x Configurator 82
11.3 Properties to Prevent Cross-Site Script Vulnerabilities 83
11.4 Setting the Database Connection Properties 84
11.5 Setting the Data Migrator Properties 85
11.6 Properties for LDAP User Authentication 86
11.7 Properties for Logging 88
11.8 Properties for Integration and Agent Servers 90

12 Configuring Yantra 7x Utilities

12.1 Installation Utilities 94
12.1.1 Loading the Yantra 7x Database Factory Defaults 95
12.1.2 Verifying the Database 97
12.2 Upgrade Utilities 98
12.2.1 Data Migrator 99
12.2.1.1 Data Migrator Task Definition Files 99
12.2.1.2 Data Migrator Restart File 100
12.2.1.3 Data Migrator Log Files 100
12.3 Development Utilities 101
12.4 Runtime Utilities 101
12.4.1 Setting Up the Classpath for the Runtime utilities: 102

12.5 Production Utilities 104
12.6 Copying a WMS Node Configuration to a New Node 104

13 Deploying Yantra 7x

13.1 Yantra 7x Enterprise Archive Package 106
13.1.1 Build Targets 108
13.2 Deploying Custom Classes 109
13.3 Support for Mixed (Secure and Unsecure) Protocols in the Yantra 7x Application Consoles 110
13.3.1 Configuring and Enabling the Filter 110
13.3.2 Mechanics of the filter 113
13.3.3 Securing Login Information When Using Mixed Protocols 114
13.4 Setting Up Scripts for Creating Yantra 7x EAR 114
13.5 Creating Yantra 7x EAR 115
13.6 Deploying Yantra 7x EAR 115
13.7 Deploying Yantra 7x on WebLogic 115
13.7.1 Setting Up the WebLogic Application Server 116
13.7.1.1 Preparing Your WebLogic Setup for JNDI Cleanup 118
13.7.2 Configuring WebLogic for Yantra 7x 119
13.7.2.1 Disabling Instrumented Stack Traces in WebLogic 119
13.7.2.2 Setting up WebLogic to Display Barcodes and Graphs 120
13.7.3 Setting up WebLogic to Use WebServices 120
13.7.4 Creating and Deploying the Enterprise Archive on WebLogic 123
13.7.5 Setting Up WebLogic to Use HTTP In-Memory Session Replication 125
13.8 Deploying Yantra 7x on WebSphere 125
13.8.1 Preparing WebSphere for Yantra 7x 126
13.8.2 Configuring WebSphere JVM Settings 126
13.8.3 Configuring a WebSphere Virtual Host for Yantra 7x 128
13.8.4 Setting Up WebSphere to Display Barcodes and Graphs 128
13.8.5 Creating the Enterprise Archive on WebSphere 129
13.8.6 Configuring the Enterprise Archive on WebSphere 132
13.8.6.1 Importing the EAR 132
13.8.6.2 Adding Resource References to an EJB Module 132
13.8.6.3 Adding Resource References to an WAR Module 133
13.8.6.4 Exporting the EAR for Deployment 134

13.8.7 Deploying the Enterprise Archive in WebSphere 134
13.8.8 Application Clients Invoking Yantra 7x EJBs 135
13.9 Setting the Client Character Display 136
13.10 Clearing Browser and Java Plugin Caches 136
13.11 Verifying Your Yantra 7x Deployment 137
13.12 Statistics Monitoring 137

14 Deploying Configuration Data

14.1 Concepts Regarding Configuration Data Deployment 140
14.1.1 Source and Target Environments 141
14.1.2 Configuration Groups and Driver Entities 141
14.1.3 Externally Maintained Configuration Data 142
14.1.4 Deploying Database Extensions 143
14.1.5 Deploying Custom Tables 144
14.1.6 Foreign Key Checks 144
14.1.7 Data Transformations 145
14.2 Before You Begin 145
14.2.1 System Requirements 145
14.2.2 Security Strategy 146
14.2.3 Change Management Strategy 147
14.2.4 Rollback Strategy 147
14.2.5 Upgrades and Maintenance 147
14.2.6 Externally Maintained Configuration Data 149
14.2.7 Best Practices 149
14.3 Setting Up the Configuration Deployment Tool 150
14.4 Running the Configuration Deployment Tool 153
14.5 Understanding the Configuration Deployment Tool User Interface 156
14.5.1 The Deployment Explorer 156
14.5.2 Comparison Results Window 157
14.5.3 The Status Panel 158
14.6 Specifying the Preferences Settings 158
14.7 Transforming Elements 160
14.8 Comparing Data 162
14.9 Examining Differences Between Databases 164
14.10 Exporting Comparison Results 165

14.11 Importing Configuration Differences....166
14.12 Generating a Report of Differences....167
14.13 Deploying Your Configuration Data....168
14.14 Deploying Your Configuration Data in Command-Line Mode....169
14.15 Troubleshooting....171
14.15.1 Informational Messages....171
14.15.2 Warning Messages....171
14.15.3 Unexpected Errors....173
14.15.4 Data Rollback Scripts....174
14.15.4.1 Customizing the Scripts....174
14.15.4.2 Running the Scripts....175

A Product Licensing and Packaging

A.1 Yantra 7x Software License Agreement....177
A.2 Yantra 7x Components....178
A.3 Yantra 7x Platform....179
A.4 Yantra 7x Applications....179
A.4.1 Yantra 7x Distributed Order Management....180
A.4.2 Yantra 7x Supply Collaboration....180
A.4.3 Yantra 7x Logistics Management....181
A.4.4 Yantra 7x Reverse Logistics....182
A.4.5 Yantra 7x Inventory Synchronization....182
A.4.6 Yantra 7x Delivery and Service Scheduling....183
A.4.7 Yantra 7x Warehouse Management System....183
A.4.7.1 WMS Inbound....184
A.4.7.2 WMS Inventory Management....184
A.4.7.3 WMS Value Added Services (VAS)....185
A.4.7.4 WMS Outbound....185
A.4.7.5 Yantra 7x Carrier Server (YCS)....185
A.4.8 Third-Party Licensing....186

Index

Preface

This manual explains how to install the Yantra 7x components and contains information relevant to new installs and upgrades of the Yantra 7x application. It also describes the major tools and components of Yantra 7x and provides information on how to set them up in a typical installation.

Intended Audience

This manual provides installation and administration information for individuals responsible for installing and maintaining Yantra 7x.

Structure

This document contains the following sections:

Chapter 1, "Getting Started"

This chapter provides a high-level introduction to the tasks involved in installing Yantra 7x.

Chapter 2, "System Requirements"

This chapter describes the hardware and software requirements for installing Yantra 7x.

Chapter 3, "Creating a Security Plan"

This chapter provides security recommendations and guidelines.

Chapter 4, "Installing Yantra 7x"

This chapter provides step-by-step instructions for installing the Yantra 7x application suite on each of the supported operating systems.

Chapter 5, "Installing the Yantra 7x Language Pack"

This chapter provides step-by-step instructions for installing the Yantra 7x language pack on each of the supported operating systems.

Chapter 6, "Installing and Configuring Application Tier Software"

This chapter provides information directing you to installation instructions for specific application server software.

Chapter 7, "Installing and Configuring Database Tier Software"

This chapter provides step-by-step instructions for installing and configuring your database server to run Yantra 7x.

Chapter 8, "Installing the Print Server"

This chapter describes the Yantra 7x WMS specific settings for the installation and configuration of the Loftware Label Manager (LLM) and Loftware Print Server (LPS).

Chapter 9, "Installing the Weighing Scale"

This chapter describes how to install the weighing scale software used by Yantra 7x WMS.

Chapter 10, "Installing the Yantra 7x Mobile Application"

This chapter describes how to install the Yantra 7x Mobile Application on PocketPC, WinCE, and VT220 mobile devices.

Chapter 11, "Configuring Yantra 7x Properties"

This chapter describes how to configure your Yantra 7x implementation after installation for minimal operations and optional components such as an LDAP server for user authentication.

Chapter 12, "Configuring Yantra 7x Utilities"

This chapter describes how to configure your Yantra 7x utilities such as the installation, runtime, and migration script files.

Chapter 13, "Deploying Yantra 7x"

This chapter describes how to deploy Yantra 7x on an application server.

Chapter 14, "Deploying Configuration Data"

This chapter describes how and when to use the Configuration Deployment Tool, which supports the day-to-day migration of Yantra 7x configuration data between two Yantra 7x environments.

Appendix A, "Product Licensing and Packaging"

This appendix provides the Yantra 7x licensing agreement, a brief description of each of the product components, a description of each of the Yantra applications, and the user interface and functional components licensed with each application.

Yantra 7x Documentation

For more information about the Yantra® 7x components, see the following manuals in the Yantra® 7x documentation set:

- *Yantra® 7x Release Notes*
- *Yantra® 7x Installation Guide*
- *Yantra® 7x Upgrade Guide*
- *Yantra® 7x Performance Management Guide*
- *Yantra® 7x High Availability Guide*
- *Yantra® 7x System Management Guide*
- *Yantra® 7x Localization Guide*
- *Yantra® 7x Customization Guide*
- *Yantra® 7x Integration Guide*
- *Yantra® 7x Product Concepts*
- *Yantra® 7x Warehouse Management System Concepts Guide*
- *Yantra® 7x Platform Configuration Guide*
- *Yantra® 7x Distributed Order Management Configuration Guide*
- *Yantra® 7x Supply Collaboration Configuration Guide*
- *Yantra® 7x Inventory Synchronization Configuration Guide*

- *Yantra® 7x Product Management Configuration Guide*
- *Yantra® 7x Logistics Management Configuration Guide*
- *Yantra® 7x Reverse Logistics Configuration Guide*
- *Yantra® 7x Warehouse Management System Configuration Guide*
- *Yantra® 7x Platform User Guide*
- *Yantra® 7x Distributed Order Management User Guide*
- *Yantra® 7x Supply Collaboration User Guide*
- *Yantra® 7x Inventory Synchronization User Guide*
- *Yantra® 7x Logistics Management User Guide*
- *Yantra® 7x Reverse Logistics User Guide*
- *Yantra® 7x Warehouse Management System User Guide*
- *Yantra® 7x Mobile Application User Guide*
- *Yantra® 7x Analytics Guide*
- *Yantra® 7x Javadocs*
- *Yantra® 7x Glossary*
- *Yantra® 7x Carrier Server Guide*
- *Yantra® 7x Application Server Installation Guide* (for optional component)

Conventions

In this manual, Windows refers to all supported Windows operating systems.

The following conventions may be used in this manual:

Convention	Meaning
. . .	An ellipsis represents information that has been omitted.
< >	Angle brackets indicate user-supplied input.
`mono-spaced text`	Mono-spaced text indicates a file name, an API name, or a code example.

Convention	Meaning
/ or \	Slashes and backslashes are file separators for Windows, UNIX and LINUX operating systems. The file separator for the Windows operating system is "\" and the file separator for Unix and Linux systems is "/". The Unix convention is used unless otherwise mentioned.

1

Getting Started

This chapter provides a high-level introduction and checklist for the tasks involved in installing Yantra 7x.

1.1 Before You Begin

Before you begin installing Yantra 7x, read this guide thoroughly. Then define your processes for handling the following:

- Development and Test Environments
- Security Strategy
- Change Management Strategy
- Development and Test Procedures
- Rollback Strategy
- Upgrades and Maintenance Strategy

In addition, before beginning the installation process, read the *Yantra 7x Performance Management Guide* which contains information that helps you optimize the performance of your Yantra 7x system.

1.2 Installation Checklist

When installing the components used by Yantra 7x follow the sequence of tasks provided in the "Checklist" and additional instructions in the chapters of this guide.

During the installation and set up processes, you should also frequently refer to the *Yantra 7x Performance Management Guide* which is a companion guide and should be used during each step of the process.

Doing so can eliminate future problems and help you to troubleshoot errors.

Checklist

1. Ensure that you have necessary system requirements to install and run Yantra 7x.
2. Set up your security infrastructure.
3. Install the Yantra 7x application.
4. Install the Yantra 7x language pack (optional).
5. Install and configure your application server.
6. Install and configure your WebServer or Proxy Server.
7. Install and configure your database software.
8. Size your database.
9. Install the print server.
10. Install the weighing scale software.
11. Install the Yantra 7x Mobile application.
12. Configure the Yantra 7x properties to use with the database, agent servers, LDAP servers, logging, and so on.
13. Configure the Yantra 7x utilities for installation, runtime, migration, and production.
14. Set up the application server for use with Yantra 7x .
15. Build your Enterprise Archive (EAR).
16. Deploy the EAR to your application server as appropriate.
17. Optionally run the configuration deployment tool to migrate your configuration data.

2

System Requirements

Yantra 7x is an n-tier application, using a combination of application server, web server, and database server software. This chapter lists all supported operating systems and the required software used in the deployment of Yantra 7x. Before installing Yantra 7x, verify that you have already installed the applicable software listed in this chapter.

This chapter provides the information required to complete step 1 of the installation checklist; as indicated by the highlighted entry:

Checklist	
1.	Ensure that you have necessary system requirements to install and run Yantra 7x.
2.	Set up your security infrastructure.
3.	Install the Yantra 7x application.
4.	Install the Yantra 7x language pack (optional).
5.	Install and configure your application server.
6.	Install and configure your WebServer or Proxy Server.
7.	Install and configure your database software.
8.	Size your database.
9.	Install the print server.
10.	Install the weighing scale software.
11.	Install the Yantra 7x Mobile application.
12.	Configure the Yantra 7x properties to use with the database, agent servers, LDAP servers, logging, and so on.
13.	Configure the Yantra 7x utilities for installation, runtime, migration, and production.

Checklist

14. Set up the application server for use with Yantra 7x .
15. Build your Enterprise Archive (EAR).
16. Deploy the EAR to your application server as appropriate.
17. Optionally run the configuration deployment tool to migrate your configuration data.

Minimum Requirements

This chapter describes the *minimum* supported options, for which Yantra 7x is already tuned for optimal performance. Your own results are derived from your specific hardware, data volumes, and user activities.

Obtaining Maximum Performance

For information on how to obtain *maximum* performance of Yantra 7x on this supported hardware and software, you must thoroughly read, evaluate, and apply any relevant recommendations described in the *Yantra 7x Performance Management Guide*. For example, Yantra 7x predefines a set of indices but also expects your Database Administrator to monitor the system and add or remove indices as necessary.

Note: You cannot remove any *unique* indices that are provided by Yantra 7x in order to avoid data integrity issues.

2.1 Technical Stack Matrix

The Yantra 7x technical stack consists of the various tiered hardware and software required by the Yantra 7x application suite. The technical stack consists of a specific application server, JDK, and database server. You can select various supported configurations from the matrix to create a supported technical stack by following these rules and instructions:

- Select one Database Server.
- Select one Application Server.
- Find a match for the operating system based on the your selections of the database and application server.

For example, you may use an Oracle 10.2.0.1 database server on a Solaris 2.9 operating system with WebSphere on Linux Red Hat 3.0.

- Although heterogeneous stacks are supported, when possible, one should consider a homogeneous stack to eliminate having to manage multiple vendors.
- MS SQL Server is supported as a low volume platform with less than 20,000 order lines that can be processed at peak hours.
- Linux is supported only on Intel processors.
- RedHat is the only Linux vendor supported.
- Oracle RAC is supported only on 2-node configurations.

2.2 Database Tier Requirements

This section describes the minimum supported options for the database tier supported in Yantra 7x applications. You can select one database server based on your choice of operating system as shown in Table 2–1, "Supported Database Tier".

Table 2–1 Supported Database Tier

Database Version	Operating System
Oracle 10.2.0.1	HP-UX 11.22 (11i V2)
	HP-UX 11.22 on Itanium
	IBM AIX 5.3 ML4
	Sun Solaris 2.9
	Redhat Linux 3.0 ES or AS on Intel 32 or 64 bit Xeon processor
	Windows 2003
DB2 8.2.3	IBM AIX 5.3 ML4
MS SQL Server 2000 SP4 with Opta 6.05 driver	Windows 2003 The Opta driver can be obtained from http://www.inetsoftware.de.
Note: Oracle RAC is supported only on 2-node configurations.	

2.3 Application Server Requirements

This section describes the minimum supported requirements for the agent and application server tier. You can select one application server based on your choice of operating system and Java Messaging Service (JMS) as shown in Table 2–2, "Supported Agent and Application Server Tier".

Note: Install the 1.4.2.* JDK that is shipped with your application server, unless otherwise noted.

Table 2–2 Supported Agent and Application Server Tier

Application Server	Operating System	JMS	JDK
BEA WebLogic 8.1 SP5	HP-UX 11.22	BEA JMS WebLogic 8.1 SP5	HP JDK1.4.2.08
	HP-UX 11.22 on Itanium	BEA JMS WebLogic 8.1 SP5	HP JDK1.4.2.08
	IBM AIX 5.3 ML4	BEA JMS WebLogic 8.1 SP5	IBM JDK1.4.2 (32 bit)
	Sun Solaris 2.9	BEA JMS WebLogic 8.1 SP5	Sun JDK1.4.2_08
	Redhat Linux 3.0 ES or AS on Intel 32 or 64 bit Xeon processor	BEA JMS WebLogic 8.1 SP5	JRockit 1.4.2_08 shipped with your WebLogic server.
	Windows 2003	BEA JMS WebLogic 8.1 SP5	Sun JDK1.4.2_08
IBM WebSphere 6.0.2.7	IBM AIX 5.3 ML4	IBM WebSphere MQ 6.0	IBM JDK1.4.2 (32 bit)
	Sun Solaris 2.9	IBM WebSphere MQ 6.0	Sun JDK1.4.2
	Redhat Linux 3.0 ES or AS on Intel 32 or 64 bit Xeon processor	IBM WebSphere MQ 6.0	IBM JDK1.4.2 cxia32412-20050609

You can install the web server or proxy server that is specified by the application server you choose, to use load balancing and achieve failover.

2.4 Internet Browser and Plugin Requirements

The minimum requirements for the internet browser and java plugin are shown in Table 2–3, "Supported Browser and Yantra 7x Client".

Table 2–3 Supported Browser and Yantra 7x Client

Internet Browser	Operating System	JRE
MS Internet Explorer 6.0 SP2	• Windows 2000, • Windows XP or • Windows 2003	Java Plugin 1.4.2_03

Note: Set the java plugin memory to 256M when using the Fulfillment Network Model. For more information on the Fulfillment Network Model see the *Yantra 7x Distributed Order Management Configuration Guide*.

2.5 Third-Party Software Requirements

The requirements for third-party systems such as Cognos Reports, Loftware Print Server, weighing scale, and so forth are provided in Table 2–4, "Supported Third-Party Software". The software mentioned in the table is supported for all of the operating systems unless otherwise noted.

Table 2–4 Supported Third-Party Software

Name	Version
Build tools	ANT 1.6.5, ANT-CONTRIB (bundled with Yantra 7x)
Analytics Reports	Cognos ReportNet 1.1 MR2
Analytics Cubes	Cognos 7.3 MR1 (Windows only)
Connectship	ConnectShip version 5.3
FedEx	FXRS 0723
FedEx Printer	Eltron LP2844
Jasper Reports	jasperreports-1.0.0.jar
Loftware Print Server	Loftware 8.3.2
Weighing Scale	Mettler-Toledo PS30, PS60, or equivalent.

2.6 Yantra 7x Mobile Application Requirements

The minimum system requirements supported for installing the Yantra 7x Mobile Application are shown in Table 2–5, "Supported Devices to Run the Yantra 7x Mobile Application".

Table 2–5 Supported Devices to Run the Yantra 7x Mobile Application

Terminals	Supported
Mobile Terminals	Pocket PC 2003 OS
	Windows CE 4.1 OS and above
	VT220 emulation software
ncurses (VT220 emulation software)	Version 5.3
ncurses build utilities	`gcc-3.3.2, bison-1.875, make-3.80, and flex-2.5.4a`. These utilties are available in the binary format at http://hpux.cs.utah.edu.
Note: Yantra 7x is specifically tested with Symbol PPT8846, Symbol PDT8146, and Symbol VRC7900 series.	

The device requirements such as memory, screen resolution, keys, and network connectivity are described in Table 2–6, "Device Requirements".

Table 2–6 Device Requirements

Options	Description
Keys	Space, Backspace, Tab, Enter, arrow keys, 0-9, A-Z, a-z, function keys (F1-F12), and special symbols such as !@#$%^&*()-_+=[]\<>?/.,. The special symbols are required only if the data (for fields such as Item ID, Location ID, and Shipment Number) contains special characters.
Barcode Scanner	This device should be equipped with an integrated barcode scanner and should have the ability to send a TAB after the scanned data.
Note: Microsoft ActiveSync 3.7 or above is required to synchronize the PC with the mobile terminal to run the Yantra 7x Mobile Application. Microsoft ActiveSync 3.7 can be installed on any PC. For more information about Microsoft ActiveSync, and related system requirements go to http://www.microsoft.com.	

Table 2–6 Device Requirements

Options	Description
Screen Resolution	Width:240, Height:320. Screens are designed for this resolution. Screen performance on devices with other resolutions may be sub-optimal. For VT220 screens, the Yantra 7x application assumes 8 rows by 20 columns.
Network Connectivity	802.11b
Memory	Minimum of 32MB RAM.
Note: Microsoft ActiveSync 3.7 or above is required to synchronize the PC with the mobile terminal to run the Yantra 7x Mobile Application. Microsoft ActiveSync 3.7 can be installed on any PC. For more information about Microsoft ActiveSync, and related system requirements go to http://www.microsoft.com.	

3

Creating a Security Plan

This chapter provides security recommendations and guidelines for running the Yantra 7x application. It is intended to help you create a reasonably secure implementation of the application.

This chapter provides the information required to complete step 2 of the installation checklist; as indicated by the highlighted entry:

Checklist	
1.	Ensure that you have necessary system requirements to install and run Yantra 7x.
2.	Set up your security infrastructure.
3.	Install the Yantra 7x application.
4.	Install the Yantra 7x language pack (optional).
5.	Install and configure your application server.
6.	Install and configure your WebServer or Proxy Server.
7.	Install and configure your database software.
8.	Size your database.
9.	Install the print server.
10.	Install the weighing scale software.
11.	Install the Yantra 7x Mobile application.
12.	Configure the Yantra 7x properties to use with the database, agent servers, LDAP servers, logging, and so on.
13.	Configure the Yantra 7x utilities for installation, runtime, migration, and production.
14.	Set up the application server for use with Yantra 7x .

Checklist

15. Build your Enterprise Archive (EAR).
16. Deploy the EAR to your application server as appropriate.
17. Optionally run the configuration deployment tool to migrate your configuration data.

Because we recognize that you may have unique business or operational requirements, Yantra cannot provide a specific set of instructions you can follow to completion for creating a security plan. Typically, it is not possible to configure a system solely for security at the detriment of other engineering or business realities.

Yantra strongly recommends that you also refer to the following documents:

- The Rhino9 Team, *The Modern Hackers Desk Reference*; available from http://www.f4.ca/text/mhdr.html.
- Tom Bialaski and Michael Haines, *Solaris and LDAP Naming Services, Deploying LDAP in the Enterprise*; Prentice Hall PTR, 2001.

3.1 Planning Your Deployment Architecture

Prior to procuring and implementing the hardware and software that make up the Yantra 7x application suite, you need to plan your deployment architecture by completing the following tasks:

- Conduct an analysis of the current security infrastructure in your organization. For more information on identifying the correct security infrastructure in the Yantra 7x application suite, refer to Section 3.1.1, "Current Security Infrastructure Analysis" on page 13.
- Conduct an analysis of authentication and authorization mechanisms in your organization to identify the steps needed to incorporate them into the Yantra 7x application suite. For more information on the mechanism used for authentication in the Yantra 7x application suite, refer to Section 3.1.2, "Authentication and Authorization" on page 13.
- Conduct an analysis of your data encryption mechanisms for deploying Yantra 7x over the internet. For more information on the different variations of the data encryption mechanisms, refer to Section 3.1.3, "Data Encryption" on page 14.

- Conduct an analysis of your organization's network topology required to deploy Yantra 7x. For more information on the various methods to deploy the application, refer to Section 3.1.4, "Network Topology" on page 14.

Completing these tasks enables you to:

- Estimate your server requirements.
- List the major security software and hardware needed to implement the Yantra 7x application.

3.1.1 Current Security Infrastructure Analysis

In order to ensure that your Yantra 7x application is a secure web application there are many factors involved. Be sure to answer the following questions before you start your Yantra 7x implementation.

- Does your organization have security personnel? If not, you may wish to seek input from an Internet security company in your area.
- Do you own a network scanner such as Internet Security Systems System Scanner or Internet Scanner? Products like these help you identify common problems with servers that are exposed to the Internet.
- Do you own an intrusion detection system such as Symantec Intruder Alert? This type of product works with your firewall to stop an intrusion before mission-critical data or systems are tampered with.

3.1.2 Authentication and Authorization

Authentication and authorization are vital to security. Due to the constantly changing authentication methodologies including biometrics, public key infrastructure (PKI), and ever-increasing encryption algorithms, Yantra provides documentation on implementing a lightweight directory access protocol (LDAP) or any Java Authentication and Authorization Service (JAAS) compliant security module for authentication. With LDAP user and password management can be centralized. For information on deploying the Yantra 7x application, and integrating with LDAP, see Section 11.6, "Properties for LDAP User Authentication" on page 86. The default authentication mechanism is implemented against the Yantra 7x database.

3.1.3 Data Encryption

Due to the differences in the nature of businesses, you may implement different security measures when implementing a web application. How you plan to deploy the application and what security measures are taken are unique to each business. Most security measures come at a cost of performance. The Internet is a public network. Sensitive data should be encrypted while traveling across it. Encrypting information that travels across the Internet has an associated cost. If the Yantra 7x application is not to be deployed on the Internet then encryption may not be necessary and the cost is thereby negated.

The data encryption mechanism recommended for Yantra 7x application is:

- SSL - 128-bit encryption is the recommended encryption level.
- VPN - 3DES is the recommended encryption algorithm.

3.1.4 Network Topology

Where is the Yantra 7x application being deployed?

- Internet?
- Virtual private network (VPN)?
- Local area network (LAN)?

3.1.4.1 Deployment Over the Internet

If you are deploying the Yantra 7x application over the Internet, you must first understand how much of your data is considered confidential. In most cases all application data about customers, vendors, and so forth is considered confidential and requires encryption. If all of the data being passed through the Yantra 7x application is to be passed through a secure socket layer (SSL), you may wish to consider purchasing an off board SSL Accelerator to offset the inherent run-time cost of encrypting information. Yantra recommends deploying 128-bit encryption.

Purchasing an SSL Accelerator can be very cost effective. For example, the run-time cost of 40-bit encryption costs 25% or more CPU utilization. What this translates to depends on your configuration and initial sizing. If your application server initial sizing was for a 4 CPU machine, and you expected to deploy 40-bit encryption, you would need one additional

processor. This entails purchasing an additional processor for your application server and an additional license for that CPU. The cost of that is typically higher than purchasing an SSL Accelerator. Yantra tests 128-bit encryption and releases the percent increase as it becomes available.

3.1.4.2 Deployment Over a Virtual Private Network

If you are deploying the Yantra 7x application over a virtual private network (VPN), the major factor in security and performance is the VPN encryption. Many firewall providers offer encryption and decryption accelerators that can be added directly to their firewalls. Checkpoint's FireWall-1, VPN-1 Accelerator Card II, is an example of this. However, one consideration for purchasing accelerator cards is how many VPN tunnels are needed. You also need to determine if the VPN is being set up for site-to-site implementation or if each individual user opens their own tunnel. If you decide on a site-to-site VPN, typically memory in the firewall is the greatest concern. If each user opens their own tunnel, processor speed is the largest concern.

In many cases the deciding factor is the speed at which your VPN is connected. If you have a T1 line, a single processor machine may suit your needs. If you plan to deploy over a T3 line, you may wish to consider a multiple-processor machine. Most firewall and VPN vendors can help you size the machine you purchase from them for optimal security and performance.

3.1.4.3 Deployment Over a Local Area Network

If you are deploying the Yantra 7x application over a local area network (LAN), there may not be a performance or security trade-off. You merely need to determine whether or not the data traveling across the LAN needs to be encrypted. If it does, then the encryption method needs to be defined.

3.2 Java Protocol Security Measures

As with the usage of any protocol technology there are certain associated risks. Yantra 7x APIs are exposed over various protocols. Therefore, Yantra strongly recommends that you disable protocols that you do not use.

3.2.1 Disabling Java Protocols

Each of the following sections provide instructions to disable the respective protocols not used.

EJB

To disable Enterprise JavaBeans (EJB) from the Yantra 7x application, comment out the “session” element in the XML descriptor file, `$YFS_HOME/descriptors/ejb-jar.xml`.

HTTP

To disable Hypertext Transfer Protocol (HTTP) as the means to enter API information in the Yantra 7x application, the deployment descriptor needs to be modified. The deployment descriptor, `web.xml`, is defined by the servlet specification from Sun Microsystems. This deployment descriptor can be used to deploy a web application on any J2EE-compliant application server. The deployment descriptor for the Yantra 7x application is stored in the `<YFS_HOME>/descriptors/<App_Server>/WAR/WEB-INF/` directory. The deployment descriptor for the InteropHttpServlet needs to be removed from the `web.xml` file to disable the servlet. Remember to remove both the `servlet-name` and the `servlet-mapping` entries from this file.

JMS

In order to use the Java Messaging Service (JMS) features of the Yantra 7x application, there must be a JMS server. There must be queues set up both on the JMS Server and within the Yantra 7x application.

To ensure that JMS is not used without authorization there should be appropriate permissions on the JMS server and in the Yantra 7x application. You may limit the ability of users to enable JMS by disabling permissions using Process Modeling in the Yantra 7x Configurator. For more information about enabling and disabling permissions, see the *Yantra 7x Platform Configuration Guide*.

3.2.2 Securing Java Protocols

Protocols are specified in the `yifclient.properties` file. To secure the protocols that you use, copy the `<YFS_HOME>/resources/yifclient.properties.sample` file and rename

to `yifclient.properties`. Ensure that this file contains the phrase `yif.apifactory.protocol=LOCAL`.

EJB

When Yantra 7x's APIs are deployed through EJB, they use a Java Naming and Directory Interface (JNDI) lookup for a context to call the EJB Objects. JNDI looks up a context that is a handle to the EJB Object or API. The APIs do not have authentication or authorization. However, security principal and credentials can be supplied by specifying them in the `yifclient.properties` configuration file. The server can be set up to validate the passed security credentials.

Yantra 7x's HTTP/HTTPS Interface uses JavaServer Pages (JSPs) installed on the application server and does not need access to JNDI. There are two ways to protect Yantra 7x APIs over EJB:

- WebLogic allows JNDI and remote method invocation (RMI) to be tunneled over HTTP. In your architecture there should be a proxy to inspect all requests for "Yantra" this ensures that all requests are for HTML, and not tunneled RMI or JNDI over HTTP.
- If the Yantra 7x application is deployed on WebLogic, a security realm should be set up to protect JNDI resources. This does not affect any screens that are packaged with the Yantra 7x application or any screens that extend Yantra 7x.

If the application is deployed on WebSphere, EJB method permissions should be set up. This does not affect any standard screens that are packaged with the Yantra 7x application or any custom screens you create.

> **Important:** If you try to run Yantra 7x using HTTPS, the Configurator will not open.

If a custom user interface is being built using Yantra 7x's APIs through EJB and not extending the Yantra 7x Presentation Framework, you cannot use the client wrapper supplied with the Yantra 7x application because it currently is incapable of passing credentials. This also applies to any use of the `YIFAPIFactory` class.

HTTP API Tester

The HTTP API tester is provided *only* to test APIs in development mode. Authentication and authorization are not used in this utility. If you plan to provide access to this page in production, you should secure access to it.

To secure access to the Yantra 7x `httpapitester` the deployment descriptor needs to be modified. The deployment descriptor's `web.xml` is defined by the servlet specification from Sun Microsystems. This deployment descriptor can be used to deploy a web application on any J2EE-compliant application server. The deployment descriptor for the Yantra 7x application is stored in the `<YFS_HOME>/descriptors/<App_Server>/WAR/WEB-INF/` directory. By using the security-constraint element with the web-resource-collection element, you can set up authorization to protect this page from unauthorized access. For more information about the `web.xml` deployment descriptor, see the documentation for your application server.

COM+

The extended Component Object Model (COM+) specification covers security in great detail. Any COM+ object deployed on a server complies with this standard. For information on setting up security for COM+ objects, see *The Microsoft Developers Network* article available at http://msdn.microsoft.com/library/default.asp?url=/library/en-us/cossdk/htm/pgservices_security_32ih.asp

3.3 Web Security

Yantra highly recommends that a security audit is made prior to deployment.

Yantra also recommends that you write log files to several servers. There are several applications that do this with no specific need for Yantra 7x to duplicate their efforts. Additionally, products like Symantec's Intruder Alert monitor log files for authentication failures and alert an administrator if a threshold is exceeded.

3.3.1 Post Installation Recommendations

After installation of the Yantra 7x application be sure to complete the following for ensured security:

1. Change the password of the default user (yantra).
2. Delete the database directory. It exposes the data model.
3. Yantra 7x supplies web pages that help you test your implementation while running Yantra 7x in development mode. The `yantrahttpapitester.jsp` web page should be removed from systems running in production mode. If you plan to provide access to this page in production, secure access to it as described in "HTTP API Tester" on page 18.
4. Change permissions on `$YFS_HOME/bin/migrator.*` files to non-executable.

3.3.2 Session Security

Session security is handled by the application server, and is stored in a non-persistent cookie on the client.

3.3.3 Operating System Permissions

The following files contain confidential information, such as user name and password combinations stored in clear text. These files should be secured through operating system permissions:

- `management.properties`
- `weblogic.log`
- `yfs.properties`
- `yifclient.properties`

3.3.4 Documentation

All documentation files for Yantra 7x and third-party software should be removed from any production servers.

3.3.5 Routing

Routing should not be enabled on a production web server.

3.3.6 Web Server Executables

Web servers should *not* be run as root. This ensures that if someone compromises any software associated with the deployment through a bug, they don't have root privileges to damage the server. Web servers allow you to access files on their host machines and as root any of those files can be modified for a deeper attack or deleted to make your web servers unavailable.

It is acceptable, although not recommended, to start the web server as `root`. A proxy server can be placed to accept HTTP traffic and redirect it to a port above 1024 on a UNIX system. If a proxy is not available and the web server must be started on port 80 it is necessary to start the web server as `root`. The web server then calls `setuid` to transfer root privileges to a generic unprivileged account. The web server's configuration file should allow you to specify what user it runs as. Any user may own the binary. The `setuid` bit should not be set on the web server binary.

3.4 Database Security

Set up separate accounts on the database server for installing the Yantra 7x schema and for accessing the application database.

If using an Oracle database on the production database server, the Oracle parameter `DBLINK_ENCRYPT_LOGIN` in your `init.ora` file should be set to `TRUE`. This ensures that all connections to the database are not sent as clear text.

3.4.1 Credit Card Encryption

If you want to ensure that credit card numbers are encrypted at the database level, you configure that functionality when setting Hub attributes in the Yantra 7x Configurator. When setting Hub attributes, make sure that the credit card number encrypting option is checked. For more information and specific instructions for setting up security, see the *Yantra 7x Platform Configuration Guide*.

Yantra 7x also supplies APIs and user exits to encrypt and decrypt credit card and other secure information. For more information about these APIs, user exits, and other data encryption, see the *Yantra 7x Customization Guide* and the *Yantra 7x Javadocs*.

3.5 Internet Explorer Security Settings

When using the Yantra 7x application without any customizations, you may need to set security or privacy settings for your Internet Explorer in order to obtain the best browser performance.

3.5.1 Configuring Browser Security Settings

To configure the Internet Explorer security and privacy settings:

1. From the Internet Explorer menu, select Tools > Internet Options > Security.
2. Select the Web content zone from which Yantra 7x is accessed.
3. Choose Default Level and set the security level to High.
4. Depending on the version of Internet Explorer you have installed, choose Custom Level and set your security settings according to one of the following:
   - Table 3–1, "Internet Explorer Version 6.0 Security Settings for Yantra 7x"
   - Table 3–2, "Internet Explorer Version 6.0 Privacy Settings for Yantra 7x"

Table 3–1 Internet Explorer Version 6.0 Security Settings for Yantra 7x

Internet Explorer Security Setting	Yantra 7x
ActiveX Controls and Plugins	
Download signed ActiveX controls	Prompt/Enable
Download unsigned ActiveX controls	Disable
Initialize and script ActiveX controls not marked as safe	Disable
Run ActiveX controls and plugins	Prompt/Enable
Script ActiveX controls marked as safe for scripting	Enable
Cookies	

Table 3–1 Internet Explorer Version 6.0 Security Settings for Yantra 7x

Internet Explorer Security Setting	Yantra 7x
Allow cookies that are stored on your computer	Disable (Enable only if you are using Yantra 7x Analytics)
Allow per-session cookies	Enable
Downloads	
File download	Enable
Font download	Prompt
Microsoft VM	
Java permissions	High Safety
Miscellaneous	
Access data sources across domains	Disable
Allow META REFRESH	Disable
Display mixed content	Prompt
Do not prompt for client certificate selection when no certificates or only one certificate exists	Disable
Drag and drop or copy and paste files	Prompt
Installation of desktop items	Disable
Launching programs and files in an IFRAME	Disable
Navigate sub-frames across different domains	Disable
Software channel permissions	High Safety
Submit non-encrypted form data	Prompt
Userdata persistence	Disable
Scripting	
Active scripting	Enable
Allow paste operations via script	Enable
Scripting of Java applets	Enable

Table 3–1 Internet Explorer Version 6.0 Security Settings for Yantra 7x

Internet Explorer Security Setting	Yantra 7x
User Authentication	
Logon	Prompt for user name and password

Table 3–2 Internet Explorer Version 6.0 Privacy Settings for Yantra 7x

Advanced Privacy Setting	Yantra 7x
Override automatic cookie handling	Yes
First-Party Cookies	Block
Third-Party Cookies	Block
Always allow session cookies	Yes

3.5.2 Adding Yantra 7x as a Trusted Website

You should set Yantra 7x to be recognized as a trusted website. Not doing so could cause certain pop-up windows such as date and time selection to display a status bar, thereby hiding certain action buttons.

To add Yantra 7x to the list of trusted websites:

1. In the Internet Explorer menu bar, select Tools > Internet Options. The Internet Options pop-up window is displayed.
2. In the Internet Options pop-up window, select the Security tab.
3. Click the Trusted Sites icon.
4. Click the Sites action button. The Trusted Sites pop-up window is displayed.
5. In the ‘Add this Web site to the zone’ text box, enter the server address where the Yantra 7x Application Consoles are installed. The port number does not need to be specified.
6. Uncheck the ‘Require server verification (https:) for all sites in this zone’ checkbox.

7. Click OK. This takes you back to the Internet Options pop-up window.
8. Click OK.

4

Installing Yantra 7x

This chapter explains how to install Yantra 7x on each operating system supported by Yantra. This chapter provides the information required to complete step 3 of the installation checklist; as indicated by the highlighted entry:

Checklist	
1.	Ensure that you have necessary system requirements to install and run Yantra 7x.
2.	Set up your security infrastructure.
3.	Install the Yantra 7x application.
4.	Install the Yantra 7x language pack (optional).
5.	Install and configure your application server.
6.	Install and configure your WebServer or Proxy Server.
7.	Install and configure your database software.
8.	Size your database.
9.	Install the print server.
10.	Install the weighing scale software.
11.	Install the Yantra 7x Mobile application.
12.	Configure the Yantra 7x properties to use with the database, agent servers, LDAP servers, logging, and so on.
13.	Configure the Yantra 7x utilities for installation, runtime, migration, and production.
14.	Set up the application server for use with Yantra 7x .

Checklist

15. Build your Enterprise Archive (EAR).
16. Deploy the EAR to your application server as appropriate.
17. Optionally run the configuration deployment tool to migrate your configuration data.

The Yantra 7x Carrier Server (YCS) and Yantra 7x Analytics are integrated with Yantra 7x. For more information on configuring YCS based on your needs see the *Yantra 7x Carrier Server Guide*. For more information on configuring reports see the *Yantra 7x Analytics Guide*.

4.1 Before You Begin

If you are upgrading from a prior release, see the appropriate *Yantra 7x Upgrade Guide* that applies to your implementation ***before*** continuing with the installation of Yantra 7x.

Before installing Yantra 7x, ensure that you already have installed the appropriate software listed in Chapter 2, "System Requirements" on page 3. In addition, please review the additional criteria described below.

UNIX Criteria

You can install Yantra 7x locally in an X Windows environment or remotely in a text-based console environment.

4.2 Installing Yantra 7x on UNIX and Linux

To install Yantra 7x on UNIX and Linux:

1. Insert the CD-ROM that is appropriate for your operating system into your CD-ROM drive and locate the `setup.bin` file found in the root directory.

   You can set the display environment in UNIX using the following command:

   ```
   export DISPLAY=<IP_address_of_XWindows_server>:0.0.
   ```

2. From the root directory, run the `./setup.bin` command.

3. After the installer finishes, you can check for any errors that may have occurred during the installation process by reviewing the log file in the <YFS_HOME> directory (where <YFS_HOME> points to the Yantra 7x software installation directory). To locate error notations in the file, search for the string “ERROR”, or more specifically, a string such as "Status: ERROR".
4. If you received any language CDs from Yantra, insert the CD-ROM into your CD-ROM drive, open the root directory, and run the `./setup.bin` command.
5. To access the printable Yantra 7x documentation set, insert the Yantra 7x Documentation CD into your CD-ROM drive, open the Documentation directory, and, if desired, copy the PDF files to an appropriate location.
6. Configure Yantra 7x as described in Chapter 11, "Configuring Yantra 7x Properties".

Note: After installing Yantra 7x on Linux, you must create a symbolic link for the `sh` file in `/usr/bin` folder. This file exists in `/bin` folder.

4.3 Installing Yantra 7x on Windows

To install Yantra 7x on Windows:

1. Insert the CD-ROM that is appropriate for your operating system into your CD-ROM drive and the installer automatically runs. You can also run the installer manually from the root directory’s `setup.exe` command.
2. When the installer window opens, follow the directions.

> **Note:** Use the backward slash (\) as a file separator when specifying any file paths. However, you should use the forward slash (/) file separator for specifying the paths in the <YFS_HOME>\resources files.

3. After the installer finishes, you can verify if there were any errors during the installation process by reviewing the log file in the <YFS_HOME> directory. To locate any notations of errors in this file, search for the string “error”.
4. Configure Yantra 7x as described in Chapter 11, "Configuring Yantra 7x Properties".

4.4 Installing Yantra 7x on a Remote Computer

You can install Yantra 7x onto any supported remote UNIX server. You cannot install Yantra 7x onto a remote Windows server.

To install Yantra 7x on a remote UNIX server:

1. FTP the setup.bin file from the CD-ROM to the remote UNIX server.

> **Note:** Typically, if the setup.bin file is copied from a Windows environment to a UNIX environment, the permissions are changed. To ensure that the proper permissions are set, run the following command:
>
> ```
> chmod 755 setup.bin
> ```

2. Start the installation procedure using the ./setup.bin -i console command and follow the directions in Section 4.2, "Installing Yantra 7x on UNIX and Linux" on page 26.

> **Note:** After installing Yantra 7x on a remote UNIX server, you must create a symbolic link for the sh file in /usr/bin folder. This file exists in /bin folder.

4.5 Uninstalling Yantra 7x

You can uninstall Yantra 7x by running the `Uninstall_Yantra7x` executable (`Uninstall Yantra7x.exe` in Windows) from `<YFS_HOME>/UninstallerData` directory.

5

Installing the Yantra 7x Language Pack

This chapter explains how to install, load the factory defaults, and check the import mode of the Yantra 7x language packs. This chapter also provides the information required to complete step 4 of the installation checklist; as indicated by the highlighted entry:

Checklist	
1.	Ensure that you have necessary system requirements to install and run Yantra 7x.
2.	Set up your security infrastructure.
3.	Install the Yantra 7x application.
4.	Install the Yantra 7x language pack (optional).
5.	Install and configure your application server.
6.	Install and configure your WebServer or Proxy Server.
7.	Install and configure your database software.
8.	Size your database.
9.	Install the print server.
10.	Install the weighing scale software.
11.	Install the Yantra 7x Mobile application.
12.	Configure the Yantra 7x properties to use with the database, agent servers, LDAP servers, logging, and so on.
13.	Configure the Yantra 7x utilities for installation, runtime, migration, and production.
14.	Set up the application server for use with Yantra 7x .

Checklist

15. Build your Enterprise Archive (EAR).
16. Deploy the EAR to your application server as appropriate.
17. Optionally run the configuration deployment tool to migrate your configuration data.

If you are upgrading from a prior release, see the appropriate *Yantra 7x Upgrade Guide* that applies to your implementation ***before*** continuing with the installation of the Yantra 7x language pack.

5.1 Installing the Language Pack

The instructions for installing the language packs for the operating systems supported by Yantra 7x are provided below.

Note: Before installing the language pack be sure that you have successfully installed Yantra 7x Release 7.5 SP1.

To install the Yantra 7x language pack on UNIX/LINUX:

Insert the language CDs that you received from Yantra into your CD-ROM drive and open the directory that is appropriate for your Unix operating system as follows:

- If you are using AIX - open the AIX directory and run the `./setup.bin` command.
- If you are using HP-UX - open the HP directory and run the `./setup.bin` command.
- If you are using Solaris - open the Sun directory and run the `./setup.bin` command.
- If you are using RedHat Linux - open the Linux directory and run the `./setup.bin` command.

To install the Yantra 7x language pack on Windows:

Insert the language CDs that you received from Yantra into your CD-ROM drive, open the "Win" directory, and run the `./setup.exe` command.

To install the Yantra 7x language pack on a Remote Computer:

You can install the Yantra 7x Language Packs onto any supported remote UNIX server. You cannot install the Yantra 7x Language Packs onto a remote Windows server.

If you received any language CDs from Yantra, FTP the `setup.bin` file from the appropriate operating system directory of the CD-ROM to the remote UNIX server.

Once the FTP is completed, start the installation procedure using the `./setup.bin -i console` command and follow the directions in Section 4.2, "Installing Yantra 7x on UNIX and Linux" on page 26.

5.2 Setting up Properties

You should configure the required parameters in your `yantra.properties.sample` file in the `<YFS_HOME>/bin` directory before you run the factory defaults. The parameters to be configured include:

```
YFS_HOME=<directory>
DB_DRIVER_CLASS=<JDBC driver class>
DB_URL=<url>
DB_USER=<User ID>
DB_PASSWORD=<Password>
ANT_OPTS=-ms96m -mx512m
log4j.configuration=/resources/log4jconfig.xml
yfs.install.localecode=<your locale code>
```

Save the modified file as `yantra.properties` in the same directory.

5.3 Loading the Yantra 7x Language Pack Factory Defaults

To load the language-specific factory defaults run the `loadDefaults.sh` script for Unix and Linux or the `loadDefaults.cmd` script for Windows available in the `<YFS_HOME>/bin` directory and pass the locale-specific installer file. For example, if you are loading the factory defaults for the Japanese language, run one of the following based on your operating system:

```
loadDefaults.cmd <YFS_HOME>/database/FactorySetup/install/
<language>_<country>_locale_installer.xml
```

The default locale that is shipped with the CD is ja_JP. If you want to define additional locales, see the *Yantra 7x Platform Configuration Guide* for information on setting up the locale.

For more information on the configuration steps to load the factory defaults see Section 12.1.1, "Loading the Yantra 7x Database Factory Defaults".

5.3.1 Loading the Yantra 7x Language Pack Translations

Prior to loading the Yantra 7x Language Pack factory defaults, be sure that you have successfully completed all instructions in Chapter 7, "Installing and Configuring Database Tier Software".

To load the language pack translations with custom localization literals, you should run the LocalizedStringReconciler tool in IMPORT mode from the <YFS_HOME>/bin directory as follows:

```
ant -f localizedstringreconciler.xml  import
-Dsrc=<YFS_HOME>/database/FactorySetup/XMLs
```

This tool first inserts the values specified in the <from_language>_<from_country>_ycplocalizedstrings_<to_language>_<to_country>.properties file present in the <YFS_HOME>/database/FactorySetup/XMLS/<language>_<country> directory into the database.

Important: Verify that your locale settings, such as currency, time format, date, and so forth, are correct.

5.3.2 Switching the Yantra 7x Base Language

The base language for the Yantra 7x Configurator can be switched only once. For more information on switching the base language and performing the switch test, see the *Yantra 7x Localization Guide*.

5.4 Creating and Deploying the Enterprise Archive

If you are installing both the Yantra 7x application and the language pack together, it is sufficient if you create and deploy the EAR once. If you already have deployed your application and are later installing the language pack you need to re-create and re-deploy the EAR file.

For more information on creating and deploying the EAR file for your chosen application server see Section 13.5, "Creating Yantra 7x EAR" on page 115 and Section 13.6, "Deploying Yantra 7x EAR" on page 115, respectively.

6

Installing and Configuring Application Tier Software

Before installing a application server, ensure that you have installed the required software mentioned in Chapter 2, "System Requirements", noting any recommendations supplied by the software provider and by Yantra Corporation. This chapter supplies information to help you install software on the application server and web server tier.

This chapter provides the information required to complete step 5 and 6 of the installation checklist; as indicated by the highlighted entry:

Checklist	
1.	Ensure that you have necessary system requirements to install and run Yantra 7x.
2.	Set up your security infrastructure.
3.	Install the Yantra 7x application.
4.	Install the Yantra 7x language pack (optional).
5.	Install and configure your application server.
6.	Install and configure your WebServer or Proxy Server.
7.	Install and configure your database software.
8.	Size your database.
9.	Install the print server.
10.	Install the weighing scale software.
11.	Install the Yantra 7x Mobile application.

Checklist

12. Configure the Yantra 7x properties to use with the database, agent servers, LDAP servers, logging, and so on.
13. Configure the Yantra 7x utilities for installation, runtime, migration, and production.
14. Set up the application server for use with Yantra 7x .
15. Build your Enterprise Archive (EAR).
16. Deploy the EAR to your application server as appropriate.
17. Optionally run the configuration deployment tool to migrate your configuration data.

6.1 Installing Your Application Server

Before installing your application server, check the requirements in Chapter 2, "System Requirements" to make sure you have the applicable hardware and software versions installed.

If you purchased the Yantra Application Server, see the *Yantra 7x Application Server Installation Guide*, located on the Yantra Application Server product CD-ROM disk, for installation instructions.

If you purchased BEA WebLogic directly from BEA, see the *Installing BEA WebLogic Platform*, located on the BEA WebLogic product CD-ROM disk, for installation instructions.

If you purchased WebSphere directly from IBM, see the *WebSphere Installation Guide*, located on the WebSphere product CD-ROM disk, for installation instructions.

6.2 Installing and Configuring Application Server Utilities

You need to configure certain application server utilities before installing Yantra 7x.

6.2.1 Installing JDK Upgrades

You should install the Java Development Kit (JDK) that is shipped with your application server. When upgrading the JDK, be sure to set the correct JAVE_HOME environment variable and update the PATH.

6.3 Installing and Configuring your Proxy Server

You can install a proxy or a web server to avoid any bottlenecks that might occur when systems are trying to access the Yantra 7x installed on your application server. Yantra recommends that you install and configure the web server version specified by your application server.

7

Installing and Configuring Database Tier Software

This chapter describes how to install and configure the database tier software to run Yantra 7x.

This chapter provides the information required to complete step 7 and 8 of the installation checklist; as indicated by the highlighted entry:

Checklist

1. Ensure that you have necessary system requirements to install and run Yantra 7x.
2. Set up your security infrastructure.
3. Install the Yantra 7x application.
4. Install the Yantra 7x language pack (optional).
5. Install and configure your application server.
6. Install and configure your WebServer or Proxy Server.
7. Install and configure your database software.
8. Size your database.
9. Install the print server.
10. Install the weighing scale software.
11. Install the Yantra 7x Mobile application.
12. Configure the Yantra 7x properties to use with the database, agent servers, LDAP servers, logging, and so on.
13. Configure the Yantra 7x utilities for installation, runtime, migration, and production.
14. Set up the application server for use with Yantra 7x .

Checklist

15. Build your Enterprise Archive (EAR).
16. Deploy the EAR to your application server as appropriate.
17. Optionally run the configuration deployment tool to migrate your configuration data.

Before installing your database server, verify that you have the applicable software versions. For more information see Chapter 2, "System Requirements".

7.1 Installing Oracle

You can use an Oracle database for maintaining information on Yantra 7x. The following sections provide the necessary steps to install and configure an Oracle database for production.

To install Oracle:

Follow the steps below to install Oracle with single or multiple byte characters:

1. If you do not have Oracle installed, follow the installation procedures in your Oracle Installation manuals.
2. Run the create instance procedure. Use a character set appropriate for your desired localanguage. For example, a single-byte character set typically requires UTF-8, while a multi-byte character set like Japanese, may require a UTF-16 character set.

   ```
   CHARACTER SET "UTF-8"
   ```

3. Configure the INIT<INSTANCE_NAME>.ORA file for Oracle as follows:

   ```
   open_cursors= <set to appropriate value>
   ```

   For example, the minimum value for WebLogic equals number of threads (across all application servers) + (connection pool size X prepared statement pool size)

   ```
   cursor_sharing=similar
   compatible=<10.1.0.2>
   timed_statistics=true
   db_block_size=8192
   ```

```
optimizer_mode=CHOOSE
```

If you are using multiple-byte character sets, set the following and restart Oracle:

```
nls_length_semantics=CHAR
```

Alternatively you can run:

```
alter session set nls_length_semantics = CHAR
```

prior to running any create table scripts.

Setting this attribute ensures that the field sizes are not impacted by the number of bytes a data type can store. For example, Varchar(40) would now be able to store 40 Japanese characters instead of 40/3 bytes in the UTF-8 character set.

4. Copy the Oracle JDBC driver file `<ORACLE_HOME>/jdbc/lib/ojdbc14.jar` from your database server to the `<YFS_HOME>/lib/ojdbc14.jar` file on your application server machine.

7.1.1 Oracle Database User Privileges

Unless specifically stated for a given task, the Yantra 7x user does not require database administrator (DBA) privileges.

The following are some of the basic privileges that are given to the Yantra 7x administrative user who is involved in creating and modifying the Oracle database:

- ALTER SESSION
- CREATE PROCEDURE
- CREATE SEQUENCE
- CREATE SESSION
- CREATE SYNONYM
- CREATE TABLE
- CREATE VIEW
- EXECUTE ANY PROCEDURE
- INSERT ANY TABLE

- UPDATE ANY TABLE
- SELECT ANY TABLE

The following are some of the basic privileges given to the application user whose involvement is restricted just to running the application:

- ALTER SESSION
- EXECUTE ANY PROCEDURE
- INSERT ANY TABLE
- UPDATE ANY TABLE
- SELECT ANY TABLE

7.1.2 Configuring an Oracle database for Production

You need to configure your Oracle database for running in a production environment with Yantra 7x. To configure an Oracle database for a production environment, you must:

- Size the database by estimating the required disk space.
- Run the database scripts to create tables, indexes and so forth for the Oracle database.
- Create views and db_link or synonyms for integrating with the Yantra 7x Warehouse Management System installation.
- Set the database connection properties.

Note: The Oracle schema having Yantra 7x database objects must have the "QUERY REWRITE" system privilege to create the function-based indexes provided in YFS_HOME/database/oracle/scripts/yfs_addnl_index.sql on the YFS_PERSON_INFO table.

In addition, in order for Oracle to use such function-based indexes in queries, the QUERY_REWRITE_ENABLED parameter must be set to TRUE, and the QUERY_REWRITE_INTEGRITY parameter must be set to TRUSTED.

To create the oracle database to handle multiple byte characters:

1. Do not modify Yantra 7x DDL.
2. Choose the correct data encoding format for your language. See "To install Oracle:" for more information.
3. Choose the character set suitable for your language. See "To install Oracle:" for specific settings to ensure the database field sizes.

7.1.2.1 Running Scripts for an Oracle database

To configure your Oracle database for your production environment, you must set up and run a series of scripts to create the tables, indices, sequences and so forth for your schema.

These script files reside in the <YFS_HOME>/database/oracle/scripts/ directory. The yfs_master_db_script.sql script is the master script that calls all other scripts required for creating tables, indexes, and so forth.

To set up scripts:

If you are using locally managed tablespace or another utility to size your database, complete the following:

1. Create tablespaces where the Yantra 7x tables and indexes reside.
2. Modify the yfs_tables.sql file to reference your newly created tablespaces.

The DDLs in the Yantra 7x scripts create a standard set of indexes. You may need to create additional indexes or modify existing indexes according to your business practice.

To run the scripts:

1. Log into the Oracle server manager as sysdba.
2. Create the user that is the designated schema owner.
3. Grant the privileges listed in Section 1.2, "Installation Checklist" to the newly created user.
4. Log out of the Oracle Server Manager and log back in as the newly created user.
5. From the `<YFS_HOME>/database/oracle/scripts/` directory, run the `yfs_master_db_script.sql` script. This creates the tables, indexes and sequences.

   The master script also creates the CustomDB views defined in the `<YFS_HOME>/database/oracle/scripts/CustomDBViews` directory.
6. Examine the `<YFS_HOME>/database/oracle/scripts/yfs_master_db_script.log` file for database creation errors.
7. Validate the database as described in Section 12.1.2, "Verifying the Database" on page 97.
8. Load the Yantra 7x database factory defaults as described in Section 12.1.1, "Loading the Yantra 7x Database Factory Defaults" on page 95.
9. Check for the degree of parallelism, using information from the *Yantra 7x Performance Management Guide*.

7.2 Installing DB2

You can use a DB2 database for maintaining information on Yantra 7x. The following sections provide the necessary steps to install and configure a DB2 database for production.

To install DB2:

1. If you do not have DB2 installed, follow the installation procedures in your DB2 Installation manual.

   Note: When creating the DB2 database, the appropriate codepage needs to be selected for international language characters (for example, UTF-8).

2. Copy the `<DB2_HOME>/sqllib/java/db2jcc.jar` database driver file and the `db2jcc_license_cu.jar` license file on your database server to the `<YFS_HOME>/lib/` directory on your application server machine.

   Note: Various Yantra 7x scripts, such as the one used for loading the factory defaults, specify a `DB_Driver`. The `DB_Driver` specified must include ***both*** of these JAR files.

3. You need to set the following parameter to avoid memory leaks and DB2 crashes:

   ```
   db2set DB2_NUM_CKPW_DAEMONS=0
   ```

7.2.1 DB2 Database User Privileges

The DBADM role is required for performing administrative operations in the DB2 database.

7.2.2 Configuring a DB2 Database for Production

You need to configure your DB2 database for running in a production environment with Yantra 7x. To configure a DB2 database for a production environment, you must:

- Size the database by estimating the required disk space.
- Run the database scripts to create tables, indexes and so forth for the DB2 database.
- Set the database connection properties.

7.2.2.1 Running Scripts for a DB2 Database

To configure your DB2 database for your production environment, you must set up and run a series of scripts to create the tables, indexes, sequences, and so forth for your schema.

These script files reside in the <YFS_HOME>/database/db2/scripts/ directory.

> **Note:** The yfs_tables.sql script creates tables and indexes. Certain tables require a page size of 16K. You should have a tablespace to accommodate such tables. The yfs_tables.sql script can be modified to specify tablespaces for tables, indexes, and so forth. If not, DB2 automatically places tables and indexes in the available tablespaces using its internal logic.

To run the scripts:

1. Run the yfs_tables.sql script followed by the yfs_seq_db2.sql script located in the <YFS_HOME>/database/db2/scripts/ directory, using the following DB2 command line processor utility:

   ```
   db2 -tvf <filename>
   ```

2. Once the yfs_tables.sql and yfs_seq_db2.sql scripts are executed, run the scripts located in the <YFS_HOME>/database/db2/scripts/CustomDBViews directory.

7.3 Installing Microsoft SQL Server

You can use a SQL Server database for maintaining information on Yantra 7x. If you do not have SQL Server installed, follow the installation procedures in your SQL Server Installation manual.

> **Note:** Ensure that Named Pipes & TCP/IP protocols are enabled in the network utility of the SQL Server.

To create a database, ensure that the collation property you select supports all the characters for your database.

Note: Set the SQL Server database to run in the "case-insensitive" mode.

To find the character set to be used in JDBC a URL

1. Find the character set corresponding to the chosen collation property. You can find the code page of the collation property by using the following SQL statement:

   ```
   select collationproperty ('<your collation property>', 'codepage')
   ```

2. Find the JAVA character set corresponding to the codepage to use with the JDBC URL. Refer to the FAQ page of i-net software to find the character set.
3. Copy the SQL Server JDBC driver file (`Opta.jar`) to your `<YFS_HOME>/lib/` directory.

7.3.1 SQL Server Database User Privileges

The DB_DDLADMIN role is required for creating objects in the SQL Server database.

7.3.2 Configuring a SQL Server Database for a Production Environment

You need to configure your SQL Server database for running in a production environment with Yantra 7x. To configure a SQL Server database for a production environment, you must:

- Size the database by estimating the required disk space.
- Run the database scripts to create tables, indices and so forth for the SQL Server database.
- Set the database connection properties.

7.3.2.1 Running Scripts for a SQL Server database

To run the scripts:

1. Make sure you have a SQL Server client installed on your computer.

2. From the <YFS_HOME>/database/sqlserver/scripts/ directory, run the yfssqlserver_master_db_script.cmd script, and pass the ServerName, DatabaseName, UserID, and Password parameters.

   The master script also creates the CustomDB views defined in the <YFS_HOME>/database/sqlserver/scripts/CustomDBViews directory.

3. Examine the log file for errors.

7.4 Database Sizing

Database sizing is designed to give you estimates of the database growth and to assist in planning the disk requirements. The planning of the capacity required in your company and the steps to estimate the disk size are described in Section 7.4.1, "Capacity Planning", Section 7.4.2, "Disk Estimation for the Distributed Order Management Module" and Section 7.4.3, "Disk Estimation for the Networked Warehouse Management System Module".

7.4.1 Capacity Planning

There are many factors to consider when estimating the amount of disk space that will be required for the Yantra 7x application suite. As a result, trying to consider all growth factors is impractical because the user may not know the answers to many questions that are required to do a detailed forecast. Over the years the cost of disks has dramatically decreased, and the capacity and speed of disks has increased. The method of how information system managers order disk capacity has also changed from purchasing disk arrays that are dedicated to a particular database server and project to the concept of SANS.

Yantra 7x provides a methodology to estimate your initial disk requirements. Consider the confidence that you have in your data estimates when making the final purchase decision and adjust accordingly. After the initial purchase and production deployment, disk growth should be tracked for future purchase forecasts.

If you use or are planning to use both Distributed Order Management (DOM) and networked Warehouse Management System (nWMS) modules, please use Table 7–1, "Steps for Disk Space Estimation for the Order Management Module" and Table 7–2, "Steps for Disk Space Estimation for the Networked WMS Module - If you have both DOM and

nWMS". However, if you are planning to use only the nWMS module use Table 7–3, "Steps for Disk Space Estimation for Networked WMS Module - If you have only nWMS".

7.4.2 Disk Estimation for the Distributed Order Management Module

The disk estimation provided here pertains to the Order Management module of the Yantra 7x application suite.

The estimation methodology consists of three parts:

1. Estimate the number of orders and order lines you expect to keep in the database.
2. Multiply the number obtained in Step 1 mentioned above, by a storage usage factor.
3. Finally add a minimum base amount.

However, the following information is essential to keep in mind before calculating the estimated disk space:

Note 1

You need to gather some information about the amount of time required to maintain the database, such as:

1. How long do you plan to keep data in the main transactional database before orders are purged to the transactional database?
2. How long are orders kept in the history database before they are purged?
3. Are you purchasing the storage for the first few years into the implementation?

Consider the following examples to achieve answers for the above mentioned questions.

Case 1 You need to purchase storage for the first 3 years of the implementation, and your company's data retention policy says that you have to keep data online in the main transactional database for 1 year and in the history database for another 5 years. Orders that are older than 6 years are purged from the system.

The following solution lets you achieve this goal:

If you need to purchase storage to cover the first 3 years of implementation, that storage has to be sufficient for 3 years worth of data. At the end of year 3, your database will have the data for the third year in the main transactional database while the data for the first and second years is in the history. In this example, you should enter the number 3 as the number of years worth of orders that you expect to keep in the database.

Case 2 The Yantra 7x application has been in production for 10 years and your company's data retention policy says that you have to keep data online in the main transactional database for 1 year and in the history database for another 5 years. Orders that are older than 6 years are purged from the system. Given the same data retention policy as above, how much storage is required?

At the end of the tenth year, the database will have the data for the tenth year in the main transactional database and the data for the fifth, sixth, seventh, eighth and ninth years in the history. Therefore, the database has six years (as dictated by the data retention policy) in the database. In this example, you should enter the number 6 as the number of years worth of orders that you expect to be kept in the database.

Note 2

The order discussed in Table 7–1, "Steps for Disk Space Estimation for the Order Management Module" includes sales, transfer, return, and work orders.

Note 3

This storage estimate is for work-in-progress tables that are used as part of order processing. When the orders are processed, the records in these tables can be purged from the system. These tables include the YFS_IMPORT, YFS_EXPORT, and so forth. ***You are strongly urged to aggressively purge data from these tables***.

Note 4

When procuring your storage, ensure that the storage device has at least the amount of usable space specified in Step 8 of Table 7–1, "Steps for Disk Space Estimation for the Order Management Module". This table provides an idea of the usable space for the storage device in your

company. However, the actual amount you might need to order, is a factor of Redundant Array of Inexpensive Disks (RAID) set up. This disk subsystem is composed of more than one disk drive to provide improved reliability, response time, and storage capacity.

Now that you have noted the above points you can proceed to the estimation of required disk space as outlined in Table 7–1.

Table 7–1 Steps for Disk Space Estimation for the Order Management Module

1.	Enter the number of years worth of information to be kept in the system (retention time). For example, the number of orders. For a more detailed example, refer to "Note 1" on page 51.	_____
2.	Enter the number of orders you expect to be in the system during the time period specified in Step 1. For the different types of orders refer to "Note 2" on page 52.	_____
3.	Enter the number of order lines present in a typical order.	_____
4.	Enter the number of order lines that are to be stored in the database (multiply the values provided in Step 2 and Step 3).	_____
5.	Enter the order line multiplier: Choose one of the following storage factors that most closely approximates a description of your Yantra 7x system: (a) 30 KB - This is primarily used for order management wtih very little customization. (b) 35 KB - This is primarily used for order management with moderate amount of customization.	_____
6.	Multiply the expected number of order lines from Step 4 and the storage factor from Step 5.	_____
7.	The minimum base storage requirement.	150 MB
8.	The minimum operational storage requirements for the Yantra 7x application. For more information on the storage estimate, refer to "Note 3" on page 52.	500 MB
9.	Enter the total estimated storage obtained by adding the values from Step 6, Step 7, and Step 8. For more information on the amount of usable space, refer to "Note 4" on page 52.	_____

7.4.3 Disk Estimation for the Networked Warehouse Management System Module

The disk estimation discussed in this section pertains to the networked WMS module of the Yantra 7x application suite.

This estimation methodology consists of three parts:

1. Estimate the number of shipment lines you expect to keep in the database.
2. Multiply the number obtained in Step 1 by a storage usage factor depending on the specifics of your implementation.
3. Add a minimum base amount for each warehouse or stockroom that you have defined.

If you are planning to use both Yantra 7x's DOM and nWMS modules use Table 7–2, "Steps for Disk Space Estimation for the Networked WMS Module - If you have both DOM and nWMS" or else if you are planning to use only the nWMS module use Table 7–3, "Steps for Disk Space Estimation for Networked WMS Module - If you have only nWMS".

However, the following information is essential to keep in mind before calculating the estimated disk space:

Note 1

You need to gather some information about the amount of time required to maintain the database, such as:

1. How long do you plan to keep data in the main transactional database before shipment data is purged to the transactional database?
2. How long is the shipment data kept in the history database before it is purged?
3. Are you purchasing the storage for the first few years into the implementation?

Consider the following example to achieve answers for the above mentioned questions.

Case 1 You need to purchase storage for the first 2 years of the implementation, and your company's data retention policy says that you have to keep data online in the main transactional database for 1

year and in the history database for another year. Shipments that are older than 2 years are purged from the system.

The following solution lets you achieve this goal:

If you need to purchase storage to cover the first 2 years of implementation, that storage has to be sufficient for 2 years worth of data. At the end of year 2, your database will have data from the second year in the main transactional database while the data from the first year is in the history. In this example, you should enter the number 2 as the number of years worth of shipment-related data that you expect to keep in the database.

Note 2

The shipment lines discussed in Table 7–2 and Table 7–3 include space requirements for demand-based replenishment.

Note 3

This storage estimate is for work-in-progress tables that are used as part of the shipment and receipt processing. When the shipments are processed, the records in these tables can be purged from the system. These tables include the YFS_IMPORT, YFS_EXPORT, YFS_TASK, YFS_TASK_STATUS_AUDIT, and so forth. ***You are strongly urged to aggressively purge data from these tables***.

Note 4

When procuring your storage, ensure that the storage device has at least the amount of usable space specified in the last step of Table 7–2 or Table 7–3. These tables provides an idea of the usable space for the storage device in your company. However, the actual amount you might need is a factor of Redundant Array of Inexpensive Disks (RAID) set up. This disk subsystem is composed of more than one disk drive to provide improved reliability, response time and storage capacity.

Now that you have noted the above points you can proceed to the estimation of required disk space as outlined in Table 7–2 or Table 7–3.

Table 7–2 Steps for Disk Space Estimation for the Networked WMS Module - If you have both DOM and nWMS

1.	Enter the number of years worth of information to be kept in the system (retention time). For example, the number of shipments. For a more detailed example, refer to "Note 1" on page 54.	_____
2.	Enter the number of shipment lines you expect to be in the system during the time period specified in Step 1. For the different types of shipments, refer to "Note 2" on page 55.	_____
3.	Enter the shipment line multiplier. This includes demand-based replenishment. Choose one of the following storage factors that most closely approximates a description of your Yantra 7x system: (a) 10 KB - for warehouses using no tag-controlled items, no serial tracking, no LNPNs or only Pallet LPNs, and more than 80% PARCEL shipping. (b) 12 KB - for warehouses using no tag-controlled items, no serial tracking, no LNPNs or only Pallet LPNs, and PARCEL as well as TL - LTL shipping. (c) 15 KB - for warehouses using tag-controlled items, no serial tracking, no LNPNs or only Pallet LPNs, and more than 80% TL - LTL shipping. (d) 20 KB - for warehouses using no tag-controlled items, no serial tracking, largely CASE LPNs, and more than 80% PARCEL shipping. (e) 25 KB - for warehouses using tag-controlled items, serial tracking, largely CASE LPNs, and PARCEL as well as TL - LTL shipping. (f) 30 KB - for warehouses using tag-controlled items, serial tracking, largely CASE LPNs, and more than 80% TL - LTL shipping.	_____
4.	Multiply the expected number of shipment lines from Step 2 and the storage factor from Step 3.	_____
5.	Enter the number of receipt lines you expect to be in the system during the time period specified in Step 1.	_____

Table 7–2 Steps for Disk Space Estimation for the Networked WMS Module - If you have both DOM and nWMS

6.	Enter the receipt line multiplier. Choose one of the following storage factors that most closely approximates a description of your Yantra 7x system: (a) 25 KB - for warehouses using no tag-controlled items, no serial tracking, and no LNPNs. (b) 27 KB - for warehouses using no tag-controlled items, no serial tracking, and no LNPNs or only Pallet LPNs. (c) 35 KB - for warehouses using no tag-controlled items, no serial tracking, and more than 80% CASE LPNs. (d) 40 KB - for warehouses using tag-controlled items, serial tracking, and more than 80% CASE LPNs.	_____
7.	Multiply the expected number of receipt lines from Step 5 and the storage factor from Step 6.	_____
8.	Enter the number of warehouses planned: (a) Enter the number of stores or stock rooms planned. (b) Enter the number of other warehouses planned.	 _____ _____
9.	Calculate the minimum space required for your set up based on the data in Step 8 and the minimum storage requirement given below: (a) 20 MB for each store or stock room. (b) 50 MB for each other warehouse.	 _____ _____
10.	The minimum operational storage requirements for Yantra 7x application. For more information on the storage estimates refer to the "Note 3" on page 55.	500 MB
11.	Enter the total estimated storage obtained by adding the values from Step 4, Step 7, Step 8 , Step 9 and Step 10. For more information on the amount of usable space, refer to "Note 4" on page 55.	_____
12.	Enter the value of Step 9 from Table 7–1 on page 53.	_____
13.	Enter the total estimated storage obtained by adding the values from Step 11 and Step 12. For more information on the amount of usable space, refer to "Note 4" on page 55	_____

Table 7–3 Steps for Disk Space Estimation for Networked WMS Module - If you have only nWMS

1.	Enter the number of years worth of information to be kept in the system (retention time). For example, the number of shipments. For a more detailed example, refer to "Note 1" on page 54.	_____
2.	Enter the number of shipment lines you expect to be in the system during the time period specified in Step 1. For the different types of shipments, refer to "Note 2" on page 55.	_____
3.	Enter the shipment line multiplier. This factor includes demand-based replenishment. Choose from one of the following storage factors that most closely approximates a description of your Yantra 7x system: (a) 25 KB - for warehouses using no tag-controlled items, no serial tracking, no LNPNs or only Pallet LPNs, and more than 80% PARCEL shipping. (b) 27 KB - for warehouses using no tag-controlled items, no serial tracking, no LNPNs or only Pallet LPNs, and PARCEL as well as TL - LTL shipping. (c) 30 KB - for warehouses using tag-controlled items, no serial tracking, no LNPNs or only Pallet LPNs, and more than 80% TL - LTL shipping. (d) 35 KB - for warehouses using no tag-controlled items, no serial tracking, largely CASE LPNs, and more than 80% PARCEL shipping. (e) 40 KB - for warehouses using tag-controlled items, serial tracking, largely CASE LPNs, and PARCEL as well as TL - LTL shipping. (f) 50 KB - for warehouses using tag-controlled items, serial tracking, largely CASE LPNs, and more than 80% TL - LTL shipping.	_____
4.	Multiply the expected number of shipment lines from Step 2 and the storage factor from Step 3.	_____
5.	Enter the number of receipt lines you expect to be in the system during the time period specified in Step 1.	_____

Table 7–3 Steps for Disk Space Estimation for Networked WMS Module - If you have only nWMS

6.	Enter the receipt line multiplier. Choose from one of the following storage factors that most closely approximates a description of your Yantra 7x system: (a) 25 KB - for warehouses using no tag-controlled items, no serial tracking, and no LNPNs. (b) 27 KB - for warehouses using no tag-controlled items, no serial tracking, and no LNPNs or only Pallet LPNs. (c) 35 KB - for warehouses using no tag-controlled items, no serial tracking, and more than 80% CASE LPNs. (d) 40 KB - for warehouses using tag-controlled items, serial tracking, and more than 80% CASE LPNs.	_____
7.	Multiply the expected number of receipt lines from Step 5 and the storage factor from Step 6.	_____
8.	Enter the number of warehouses planned: (a) Enter the number of stores or stock rooms planned. (b) Enter the number of other warehouses planned.	 _____ _____
9.	Calculate the minimum space required for your set up based on the data in Step 8 and the minimum storage requirement given below: (a) 20 MB for each store or stock room. (b) 50 MB for each other warehouse.	 _____ _____
10.	The minimum operational storage requirements for the Yantra 7x application. For more information on the storage estimates refer to "Note 3" on page 55.	500 MB
11.	Enter the total estimated storage obtained by adding the values from Step 4, Step 7, Step 8 , Step 9 and Step 10. For more information on the amount of usable space, refer to "Note 4" on page 55.	_____

7.4.4 Tracking and Estimating Future Disk Requirements

You should track your actual database storage usage and the number of database records regularly. Correlating these two metrics enabled you to plan your future disk requirements. Moreover, determining the average amount of space used for each order line or shipment line, enables you to accurately predict your future growth requirements.

8

Installing the Print Server

This chapter explains how to install and configure the Loftware Label Manager (LLM) and Loftware Print Server (LPS).

This chapter provides the information required to complete step 9 of the installation checklist; as indicated by the highlighted entry:

Checklist

1. Ensure that you have necessary system requirements to install and run Yantra 7x.
2. Set up your security infrastructure.
3. Install the Yantra 7x application.
4. Install the Yantra 7x language pack (optional).
5. Install and configure your application server.
6. Install and configure your WebServer or Proxy Server.
7. Install and configure your database software.
8. Size your database.
9. Install the print server.
10. Install the weighing scale software.
11. Install the Yantra 7x Mobile application.
12. Configure the Yantra 7x properties to use with the database, agent servers, LDAP servers, logging, and so on.
13. Configure the Yantra 7x utilities for installation, runtime, migration, and production.
14. Set up the application server for use with Yantra 7x .

Checklist

15. Build your Enterprise Archive (EAR).
16. Deploy the EAR to your application server as appropriate.
17. Optionally run the configuration deployment tool to migrate your configuration data.

For more information about configuring the Loftware Label Manager and Print Server, see the *Loftware Label Manager User's Guide* and the *Loftware Print Server User's Guide*.

For more information about Performance Considerations for setting up the Loftware Print Server (LPS) see the *Loftware Print Server User's Guide*.

8.1 Installation of Loftware Components

The Loftware Print Server manages bar code label print requests between applications and hundreds of networked printers. As a general guideline, you should configure a maximum of 200 printers for each Loftware Print Server you install. For more information about server requirements and installation instructions, see the *Loftware Print Server User's Guide*. Contact your Loftware support representative for additional sizing and configuration support.

The Loftware Label Manager, used for designing labels, may be installed on any compatible PC. For more information about server requirements and installation guidelines, see the *Loftware Label Manager User's Guide*.

Yantra 7x supports printing in the following modes:

- File Copy Mode
- TCP/IP Sockets Mode

The `yfs.loftware.tcpip.sockets` attribute in the `yfs.properties` file determines the mode used for printing. By default this boolean property is set to 'N' for File Copy Mode.

To configure the Loftware printing in the TCP/IP Sockets Mode, this property should be set to 'Y'.

Yantra 7x requires the following settings in the Loftware Print Server Configuration Utility:

- In Directory Set up, ensure that the 'Pass Files' option is selected.
- **When using File Copy Mode:** In Directory Set up, ensure that the 'Enable Polling (Disable Event File Trigger)' option is selected. It is recommended that the Poll Interval value is set to 500 Milliseconds.

> **Note:** In File Copy Mode, SAMBA should be configured when using a UNIX version of the application server.
>
> The Drop Directories of the printers configured in Loftware need to be mounted on to the UNIX server using SAMBA.

8.2 Define Printers on Loftware

Configure printers on Loftware using the Loftware Design 32 tool. For more information about configuring printers using the Loftware Design 32 tool, see the *Loftware Label Manager User's Guide*.

8.3 Define Printers for Yantra 7x WMS Installation

For more information about configuring printers for Yantra 7x WMS, see the *Yantra 7x Warehouse Management System Configuration Guide*.

8.4 Copying Yantra 7x WMS Standard Label Formats

Yantra 7x WMS provides Loftware Label Manager template (*.lwl) files which should be copied in the directory set up for labels using the Loftware Design 32 tool.

Also, copy the `YCP_LABEL_FIELDS.LST` file to the directory where the Loftware Label Manager has been installed. This file is available in the `Runtime/Template/Prints/Label` directory.

8.5 Installation of JasperReports

JasperReports is an open source Java reporting tool that delivers rich content on the screen, to the printer or in the format of a PDF, HTML and so forth. You can use JasperReports with Yantra 7x for printing order

reports, labels, and so forth. The installation procedure and sample files are located in the <YFS_HOME>/documentation/code_examples/jasperreports directory.

9

Installing the Weighing Scale

Weighing scales are typically used at packing or manifest stations. This chapter describes the installation of weighing scales for use with Yantra 7x WMS.

This chapter provides the information required to complete step 10 of the installation checklist; as indicated by the highlighted entry:

Checklist	
1.	Ensure that you have necessary system requirements to install and run Yantra 7x.
2.	Set up your security infrastructure.
3.	Install the Yantra 7x application.
4.	Install the Yantra 7x language pack (optional).
5.	Install and configure your application server.
6.	Install and configure your WebServer or Proxy Server.
7.	Install and configure your database software.
8.	Size your database.
9.	Install the print server.
10.	Install the weighing scale software.
11.	Install the Yantra 7x Mobile application.
12.	Configure the Yantra 7x properties to use with the database, agent servers, LDAP servers, logging, and so on.
13.	Configure the Yantra 7x utilities for installation, runtime, migration, and production.
14.	Set up the application server for use with Yantra 7x .

Checklist

15. Build your Enterprise Archive (EAR).
16. Deploy the EAR to your application server as appropriate.
17. Optionally run the configuration deployment tool to migrate your configuration data.

For more information regarding the Mettler-Toledo PS Weighing Scale, see the *Mettler-Toledo PS Weighing Scale User Guide*.

9.1 Installation of the Weighing Scale

The weighing scale is installed at each pack or manifest station requiring weighing scale integration.

To install the weighing scale, follow these steps on each client machine:

1. Launch your Internet Explorer browser.
2. In the Address bar, type `http://<hostname>:<PortId>/yantra/yfscommon/win32com.dll` and press Enter. The File Download window appears.
3. Choose Save. The Save As window appears.
4. Save the file in any directory present in the System Class path. For example, on Windows NT, go to `C:/WINNT/system32`.

For more information about setting up the weighing scale and associating it with a station, see the Equipment section of the *Yantra 7x Warehouse Management System Configuration Guide*.

For more information about system requirements, see Chapter 2, "System Requirements" on page 3.

10

Installing the Yantra 7x Mobile Application

This chapter describes how to install the Yantra 7x Mobile Application for use on PocketPC, WinCE, and VT220 mobile terminals.

This chapter provides the information required to complete step 11 of the installation checklist; as indicated by the highlighted entry:

Checklist	
1.	Ensure that you have necessary system requirements to install and run Yantra 7x.
2.	Set up your security infrastructure.
3.	Install the Yantra 7x application.
4.	Install the Yantra 7x language pack (optional).
5.	Install and configure your application server.
6.	Install and configure your WebServer or Proxy Server.
7.	Install and configure your database software.
8.	Size your database.
9.	Install the print server.
10.	Install the weighing scale software.
11.	Install the Yantra 7x Mobile application.
12.	Configure the Yantra 7x properties to use with the database, agent servers, LDAP servers, logging, and so on.
13.	Configure the Yantra 7x utilities for installation, runtime, migration, and production.
14.	Set up the application server for use with Yantra 7x .

Checklist

15. Build your Enterprise Archive (EAR).
16. Deploy the EAR to your application server as appropriate.
17. Optionally run the configuration deployment tool to migrate your configuration data.

For information on PocketPC and WinCE mobile terminal system requirements, see Chapter 2, "System Requirements" on page 3.

Important: Install the Microsoft .NET Compact Framework 1.0 SP2 (`.NET CF 1.0 SP2`) on your local PC, BEFORE installing the Yantra 7x Mobile Application. This may be downloaded from http://www.microsoft.com.

Installing the Microsoft .NET Compact Framework 1.0 SP2 creates multiple `.CAB` files on the system, for multiple operating systems and processors of the device.

10.1 Installation on Mobile Terminals

To set up the PC with the Yantra 7x Mobile Application for the Symbol Mobile Terminal, follow these steps:

1. Connect the Mobile Terminal to the PC that has ActiveSync installed. For more information, refer to the Communications section of the *Symbol Installation Guide*.

> **Note:** It is suggested that ActiveSync be used to copy the files.
>
> Alternatively, you may transfer the file over the network, if the device is already configured to access the LAN, or you may serve the .CAB file through a webserver and use the Internet Explorer browser on the device to download it.
>
> If Yantra 7x WMS is installed on a UNIX server, it may be required to copy the files from the UNIX server to the PC before launching ActiveSync.

2. Choose Start > Programs > ActiveSync on the PC.
3. Choose File > Get Connected, if not already connected.
4. Choose the Explore icon. This brings up the File Explorer for the Mobile Terminal.
5. Go to \Application folder.
6. In the File Explorer window, click the Folders icon to bring up folders in the left panel.
7. Copy the YantraMobileApp_xxx.CAB file from the <YFS_HOME>\YantraMobileApp folder on the PC to the \Application folder on the mobile terminal.

   Here, xxx refers to PPC.ARM, PPC.ARMV4, WCE4.ARMV4, or WCE4.ARMV4T.

   Ensure that you choose the .CAB file that is relevant to the operating system and processor of your mobile terminal.
8. Double-click on the YantraMobileApp_xxx.CAB file, on your mobile terminal. This installs the application on the mobile terminal.

> **Note:** This file is automatically deleted upon successful installation.

9. The sample file <Install_Folder>\YantraMobileApp\YantraHostList.xml includes application servers with Loopback, Production, QA and Test

names. Replace these with the application servers along with their IP addresses and port numbers. The application server names entered here will be listed in the Servers drop-down list to which you can get connected when you launch the Yantra 7x Mobile Application on the PocketPC or WinCE mobile terminals.

Note: This step is not valid for VT220 mobile terminal.

If you want to run the application on https, you need to configure securemode and provide secureURL attributes. The applicable values of securemode are 'all', 'loginonly', and 'none'.

If you set securemode to:

- 'all', the application will run on https.
- 'loginonly', only the login page will run on https and the rest of the application on http.
- 'none', secureURL will be ignored and the application will run on http.

Note: Servers include application servers used for production, test, and other environments, if applicable.

10. If you are using a PocketPC mobile terminal, rename the `YMAProperties.ppc.xml` file to `YMAProperties.xml`. If you are using a Symbol VRC7900 series truck mount device, rename the `YMAProperties.vrc7900.xml` file to `YMAProperties.xml`.
11. Stop ActiveSync.

For additional information on replication of the Yantra 7x Mobile Application to multiple mobile devices, see the *Symbol Device Installation* documentation.

The *Symbol Device Installation* documentation also provides additional details on re-installation.

Note: To display clear and appropriate error messages, a locale specific "System_SR_<locale>.cab" resource file is needed. For more information about the locale specific resource file, contact your PocketPC product support at http://support.microsoft.com.

10.1.1 Ensuring Re-installation on Cold Boot

On a PocketPC Mobile Terminal

To ensure that the Microsoft .NET Compact Framework and the Yantra 7x Mobile Application are installed automatically on cold boot, the following instructions must be followed as a one-time measure:

1. Install the Microsoft .NET Compact Framework on the mobile terminal, BEFORE installing the Yantra 7x Mobile Application. This may be downloaded from http://www.microsoft.com.
2. Install the Yantra 7x Mobile Application. For more information, see Section 10.1, "Installation on Mobile Terminals" on page 68.
3. Copy the Microsoft .NET Compact Framework installation CAB file from the local PC to the `\Application` folder on the mobile device.
4. Copy all the files under `<YFS_HOME>\YantraMobileApp` to the `\Application\YantraMobileApp` folder.
5. Copy the file `YantraMobileApp.lnk` from the `\Windows\Start Menu\Programs` folder to the `\Application` folder.
6. Edit the following line in the `yantra.cpy` file located in the `<YFS_HOME>\YantraMobileApp` folder:

   ```
   \Application\netcf.core.ppc3.ARM.cab >
   \Windows\startup\netcf.core.ppc3.ARM.cab
   ```

   Here, rename the `netcf.core.ppc3.ARM.cab` file to the CAB file name as appropriate for your handheld device.
7. Change all occurrences of `Program Files` to `<YFS_HOME>`, as applicable in all lines of the `yantra.cpy` file.

8. Save the modified yantra.cpy file under the \Application folder on the mobile device.

This ensures that the Microsoft .NET Compact Framework and the Yantra 7x Mobile Application are installed on cold boot. When cold booting the system, the yantra.cpy file copies the installation files to the start directory.

On a Symbol VRC7900 WinCE Mobile Terminal

Symbol VRC7900 supports installation of software during cold boot by putting the installation .cab files in the \FlashFx\.CAB folder.

All files that need to be copied to the folder <Install_Folder>\YantraMobileApp folder in the RAM file system must be placed under \FlashFx\CopyToRam\Root\<Install_Folder>\YantraMobileApp.

To ensure that the Microsoft .NET Compact Framework and Yantra 7x Mobile Application are installed automatically on cold boot, the following instructions must be followed as a one-time measure:

1. Install the Microsoft .NET Compact Framework on the mobile terminal, BEFORE installing the Yantra 7x Mobile Application. This may be downloaded from http://www.microsoft.com.
2. Install the Yantra 7x Mobile Application. For more information, see Section 10.1, "Installation on Mobile Terminals" on page 68.
3. Copy the Microsoft .NET Compact Framework installation CAB file, netcf.core.WINCE.ARMV4.cab, from the local PC to the \FlashFx\.CAB folder on the mobile device.
4. Copy all the files (including the hidden file vsd_setup.dll) under the <Install_Folder>\YantraMobileApp folder to the \FlashFx\CopyToRam\Root\<Install_Folder>\YantraMobileApp folder.
5. If a desktop shortcut has been created, copy the corresponding shortcut file (with .lnk extension) to the \FlashFx\CopyToRam\System\Desktop folder.
6. Copy the \Windows\My Company YantraMobileApp.unload file to the \FlashFx\CopyToRam\System folder.

This ensures that the Microsoft .NET Compact Framework and the Yantra 7x Mobile Application are installed on cold boot.

10.2 Installing on VT220 Mobile Terminals

This section describes how to install the Yantra 7x Mobile Application for use on VT220 mobile terminals.

The Yantra 7x Mobile Application can be accessed from any VT220 emulation terminal.

HP-UX operating systems require the installation of ncurses to enable you to change the function key sequence mapping.

10.2.1 Installing ncurses on HP-UX

The VT220 client on HP-UX requires `tic from ncurses` distribution, to prepare `TERMINFO` for a VT220.

Installing `ncurses` is a multiple step process which involves ensuring that the required build utilities are already installed. These utilities need to be installed prior to the installation of `ncurses`.

Specifically, you need `gcc`, `bison`, `make` and `flex` as minimum system requirements. For more information on the ncurses and the build utility versions see, Chapter 2, "System Requirements" on page 3.

These must be installed in the following order:

1. `bison`
2. `flex`
3. `make`
4. `gcc`

Once these underlying utilities are installed, you can compile or build `ncurses`.

10.2.2 Installing libiconv on HP-UX Itanium

The libiconv libraries for HP-UX Itanium B.11.23 need to be installed for running VT220 on the HP-UX Itanium platform.

> **Note:** Libiconv library has run-time dependencies on libgcc and gettext. Therefore, you must install these packages while installing libiconv.

10.2.3 Installing the Yantra 7x Mobile Application on VT220 Mobile Terminals

The Yantra 7x Mobile Application can be installed on AIX, HP-UX, Linux, or Solaris operating systems. The VT220 terminal emulation software is installed along with the Yantra 7x Mobile Application as described in this document for the respective operating systems.

To install the Yantra 7x Mobile Application for VT220 terminal emulation:

1. The Yantra 7x Mobile Application for VT220 terminal emulation is installed under the directory <YFS_HOME>/YantraMobileApp/vt220/<OS-name>; where <OS-name> is the applicable operating system from the following: aix, hpux, linux, or solaris. This installation location is referenced as <VT220_HOME>.
2. Grant execute permission to <VT220_HOME>/yantramobileapp and <VT220_HOME>/keyseq.
3. Set up the VT220 emulation terminal as described in "Setting Up The Terminal". Using the <VT220_HOME>/keyseq binary, verify that the keys F1 through F12 display the respective keys on the keyseq program output.

Setting Up The Terminal

When setting up a terminal for use with the Yantra 7x Mobile Application for VT220 terminal emulation, you must perform the following preliminary actions:

1. Set your environment variable TERM to vt220.
2. Under the <VT220_HOME> directory, create a directory called terminfo.
3. If running VT220 emulation on an HP-UX system, install ncurses as directed in Section 10.2.1, "Installing ncurses on HP-UX" and complete Step 4 through Step 6. Otherwise, skip to Step 7.
4. From the terminfo directory use the infocmp command to define your terminal information as:

   ```
   $<ncurses_home>/infocmp > vt220.ti,
   ```

where <ncurses_home> is the ncurses binaries installation location.

Note: If an installation location is not specified during the ncurses installation, the ncurses binaries are installed in the /usr/bin folder.

The infocmp command de-compiles the terminal information, and the resulting file can be edited to map the keystrokes observed by running keyseq.

5. You must define a TERMINFO variable (if you do not already have one) to tell the terminal where to find information on a particular terminal type. On BASH type systems, this is done using the following command:

   ```
   $ export TERMINFO=<path to some directory that contains the .ti file>
   ```

6. Now run the ncurses tic command as follows to compile your newly built terminal information file:

   ```
   $ <ncurses_home>/tic vt220.ti
   ```

   where <ncurses_home> is the ncurses binaries installation location.

   Note: If an installation location is not specified during the ncurses installation, the ncurses binaries are installed in the /usr/local/bin folder.

   The tic command places the compiled version in the appropriate place under the $TERMINFO directory.

7. To ensure that all function keys are properly mapped, use the <VT220_HOME>/keyseq program. This shows you what key sequence is returned when a key is pressed. Execute it and press the function keys when prompted.

   For example, if the F1 key is pressed and:

```
Press a key (Return to end): Key Value returned: 27
Press a key (Return to end): Key Value returned: 91
Press a key (Return to end): Key Value returned: 49
Press a key (Return to end): Key Value returned: 49
Press a key (Return to end): Key Value returned: 126
```

is printed to STDOUT. These values are decimal values.

Convert these values to their HEX equivalents. Then, using the "Hexadecimal - character" set from the `man ascii` command on UNIX, edit the `vt220.ti` file created above. Use Table 10–1 to decide which values to edit.

In the example above, the F1 key maps to kf1 (from Table 10–1). Therefore, you must change the value of kf1 in the `vt220.ti` file (generated in Step 4) to:

```
\E[11~
```

8. Repeat Step 7 for all the keys you want to map.
9. Re-compile the edited `vt220.ti` file as directed in Step 6.
10. Once you have edited and compiled your terminal information file, test the changes you made by running keyseq again. If all keys are properly defined, keyseq returns a string description and the numeric value of the key.

The most common keys and their codes in the terminal information file are in Table 10–1:

Table 10–1 Terminal Information - Common Keys and Codes

Code	Key
kcub1	Left arrow
kcuf1	Right arrow
kcuu1	Up arrow
kcud1	Down arrow
kend	END key
khome	HOME key

Table 10–1 Terminal Information - Common Keys and Codes

Code	Key
knp	PAGE DOWN key
kpp	PAGE UP key
kich1	INSERT key
kdch1	DELETE key
kf1 – kf12	F1 – F12 keys

To launch the Yantra 7x Mobile Application using the VT220 emulation terminal and access online help:

1. Set an environment variable `VT220_HOME` pointing to the folder containing the VT220 executable. This environment variable must be set in the shell from where the `yantramobileapp` executable will be invoked.
2. To launch the application, type the following command in the operating system shell:
   `yantramobileapp -i <ip_address> -p <port_number>`
3. To access help, type the following command in the operating system shell:
   `yantramobileapp -help`

11

Configuring Yantra 7x Properties

After installing the Yantra 7x application, you must set up a few property and script files in order for it run properly. This chapter describes the properties you must set up before configuring Yantra 7x to run according to your business needs.

This chapter provides the information required to complete step 12 of the installation checklist; as indicated by the highlighted entry:

Checklist	
1.	Ensure that you have necessary system requirements to install and run Yantra 7x.
2.	Set up your security infrastructure.
3.	Install the Yantra 7x application.
4.	Install the Yantra 7x language pack (optional).
5.	Install and configure your application server.
6.	Install and configure your WebServer or Proxy Server.
7.	Install and configure your database software.
8.	Size your database.
9.	Install the print server.
10.	Install the weighing scale software.
11.	Install the Yantra 7x Mobile application.
12.	Configure the Yantra 7x properties to use with the database, agent servers, LDAP servers, logging, and so on.
13.	Configure the Yantra 7x utilities for installation, runtime, migration, and production.

Checklist

14. Set up the application server for use with Yantra 7x .
15. Build your Enterprise Archive (EAR).
16. Deploy the EAR to your application server as appropriate.
17. Optionally run the configuration deployment tool to migrate your configuration data.

If you are upgrading from a prior release, see the *Yantra 7x Upgrade Guide* that applies to your implementation ***before*** continuing with the setup of Yantra 7x.

11.1 Setting Up the Properties Files

Yantra 7x supplies sample properties files that you must customize before using. The properties files are located in the `<YFS_HOME>/resources/` directory and should remain accessible through your `<classpath>/resources/` directory. The sample properties files are:

1. yfs.properties.sample
2. yifclient.properties.sample
3. management.properties.sample
4. log4jconfig.xml.sample
5. ycs.properties.sample
6. migrator.properties.sample

Some properties relay sensitive data such as user IDs and passwords, which you may want to encrypt. Any property (except for the `yfs.propertyencrypter.class` property in the `yfs.properties` file), can be encrypted as needed within the following files:

- management.properties
- ycs.properties
- yfs.properties
- yifclient.properties

For more information on the usage of encryption mechanisms, see Chapter 3, "Creating a Security Plan" on page 11. To get more information about the encrypter interface used in Yantra 7x, see the `YCPEncrypter` class in *Yantra 7x Javadocs*.

The log4jconfig.xml File

The `log4jconfig.xml` file specifies the logging parameters for the log4j utility.

The management.properties File

The `management.properties` file specifies how Yantra 7x processes (such as agent servers, integration servers, and the application server) communicate with each other. The properties this the file are used to connect to the JNDI registry to enable cached data change propagation, enable and disable traces, and enable the management of these processes through the system management console.

The migrator.properties File

The `migrator.properties` file contains information about the database XML files, log files generated while loading the factory defaults, output error messages, and any output statistics available.

The ycs.properties File

The `ycs.properties` file contains configuration information for carrier integration. For more information on this properties file see the *Yantra 7x Carrier Server Guide*.

The yfs.properties File

The `yfs.properties` file contains overall settings for Yantra 7x, such as your database connectivity properties and settings that enable your implementation to work in backward compatibility mode. Yantra supplies a sample properties file that has pre-configured settings for these and other parameters.

The yifclient.properties File

The `yifclient.properties` file contains parameters for client programs and user interface components that access the Yantra 7x APIs.

To set up a properties file:

1. Rename the relevant properties file in the <YFS_HOME>/resources/ directory, by removing the .sample extension.
2. Edit the file by uncommenting properties and adding values as needed, using the descriptions as provided in this guide and as noted within each properties file.

> **Note:** Examples in the sample file may contain backslashes ("\"). When specifying your own parameters, use slashes ("/") instead of backslashes.

3. When encrypting properties:
   a. Append the property you want to encrypt with ".encrypted". Note that you cannot encrypt the yfs.propertyencrypter.class property.
   b. Ensure that the class specified for the yfs.propertyencrypter.class property is accessible through the CLASSPATH environment variable.
   c. Implement the YCPEncrypter interface. For details about this interface, see the *Yantra 7x Javadocs*.

   All properties ending with .encrypted are automatically decrypted at runtime.

11.2 Properties for the Yantra 7x Configurator

Your Yantra 7x environment and associated business processes are set up through the Yantra 7x Configurator user interface.

The Yantra 7x Configurator runs as a Java Applet and by default it requires access to the Internet. If Internet access is not available to those using the Yantra 7x Configurator, download the Java plugin version from http://java.sun.com/j2se/1.4.2/download.html and install it locally on the computer of each Yantra 7x Configurator user.

Then change the yfs.config.java.plugin.codebase property in the yfs.properties file to point to that location as follows:

```
yfs.config.java.plugin.codebase=<network location of java plugin>
```

Important: If you run Yantra 7x using HTTPS, the Yantra 7x Configurator does not open. However, if running on WebLogic, you can run Yantra 7x using mixed protocols. For more information about using mixed protocols, see Section 13.3, "Support for Mixed (Secure and Unsecure) Protocols in the Yantra 7x Application Consoles" on page 110.

11.3 Properties to Prevent Cross-Site Script Vulnerabilities

In some cases, data to and from Yantra 7x can contain HTML characters that impact the display and the original intent of the input. In addition, data can be input that contains malicious HTML, such as commands embedded within <SCRIPT>, <OBJECT>, <APPLET>, and <EMBED> tags.

In Yantra 7x, potentially malicious data being output to the browser can be rendered harmless by implementing the provided encoding mechanism. This prevents these malicious scripts from being run by the browser.

To prevent cross-site scripting, enable the `yfs.htmlencoding.triggers` property in the `yfs.properties` file and specify the following characters:

- Greater than symbol (">")
- Less than symbol ("<")
- Right parenthesis (")")
- Right bracket ("]")
- Any other characters necessary for your specific implementation

Any presence of these characters triggers Yantra 7x to safely encode the data.

For more detailed information about malicious scripts, see the following articles:

- CERT Advisory, *Malicious HTML Tags Embedded in Client Web Requests.* Available from http://www.cert.org/advisories/CA-2000-02.html.
- CERT Advisory, *Frequently Asked Questions About Malicious Web Scripts Redirected by Web Sites*. Available from http://www.cert.org/tech_tips/malicious_code_FAQ.html.

11.4 Setting the Database Connection Properties

You can set the database connection properties by specifying the login parameters in the `<YFS_HOME>/resources/yfs.properties` file.

Edit the `yfs.properties` file and set the values of the properties as described in Table 11–1:

Table 11–1 Database Connection Property Settings

Property	Description
yfs.dblogin.dbtype	Set the type of your database to: • `yfs.dblogin.dbtype=oracle.` • `yfs.dblogin.dbtype=db2.` • `yfs.dblogin.dbtype=sqlserver.`
yfs.dblogin.driverclass	Set the database driver class to: • `oracle.jdbc.OracleDriver.` • `com.ibm.db2.jcc.DB2Driver.` • `com.inet.tds.TdsDriver.`
yfs.dblogin.jdbcurl	Set the JDBC URL to: • `jdbc:oracle:thin:@<Database Server Hostname/IPaddress>:<TNS Listener Port Number>:<Database SID>` • `jdbc:db2://<Database Server Hostname>:<Port Number>/<Database name>` • `jdbc:inetdae7:<Database Server HostName/IPaddress>:<Port Number>?<database=Database Name>&charset=<your charset>`
yfs.dblogin.userid	Specify the database user ID.

Table 11–1 Database Connection Property Settings

Property	Description
yfs.dblogin.password	Specify the password associated with the user ID for the database.
yfs.dblogin.yantraschema.name	Specify the name of Yantra 7x database schema owner.
yfs.dblogin.datasource.name	If you are using a BEA WebLogic or IBM WebSphereJDBC datasource, set the same name as the datasource name.

Note: Remember to comment the dbtype, driverclass and jdbcurl that does not pertain to your configuration.

To enable database connection pooling, create the pool, configure the data source entry for JNDI, and specify it as the datasource name. For more information on BEA WebLogic JDBC connection pooling, see the *Yantra 7x Performance Management Guide*. For connection pooling used by other application servers refer to the vendor's manual on connection pooling.

The `yfs.properties` file specifies the default properties used by Yantra 7x. You can override properties for a specific Yantra 7x process by creating an override properties file and invoking the process with `-Dyantra.override.properties.file=<your_filename>` as an option to the Java VM.

11.5 Setting the Data Migrator Properties

In addition to reading the Yantra 7x properties files, the Data Migrator reads additional values from the `migrator.properties` file. A sample `migrator.properties` file is provided.

The `migrator.properties` file contains information about the location of the following Data Migrator files:

- Input XMLs to load into the database
- Log files
- Output error messages
- Output statistics

The migrator.properties file should be accessible to the Data Migrator through the CLASSPATH. Therefore, your CLASSPATH should contain the <YFS_HOME> directory for resolving the migrator.properties file.

Table 11–2 Data Migrator Connection Property Settings

Property	Description
yfs.migrator.xml.directory	Set the factory defaults entity XML directory as <YFS_HOME>/database/FactorySetup/XMLS.
yfs.migrator.error.log.file	Set the name of the migrator error log file. For example, migrator_errors.log.
yfs.migrator.statistics.file	Set the migrator statistics log file name. For example, migrator_statistics.log.
yfs.migrator.logdir	Specify the directory for the above mentioned log files. For example, C:\Yantra7xMigratorLogs.

11.6 Properties for LDAP User Authentication

This section assumes you understand how LDAP servers work. Yantra also recommends that you read the following documents on LDAP technology:

- W. Yeong, T. Howes, and S. Kille, *RFC 1777 - Lightweight Directory Access Protocol*. March 1995. Available at http://rfc.sunsite.dk/rfc/rfc1777.html.
- Mark Wilcox, *Implementing LDAP*. Wrox Press, 1999.

By default, all authentication is performed against the Yantra 7x database. When a user enters a login ID and password, it is validated against the login ID and password that is stored in the database. This requires the administrator of the Yantra 7x system to set up login IDs and passwords for each user.

Alternatively, the Yantra 7x Application Consoles support LDAP-based user authentication. You may choose to use an LDAP server for authentication. When using LDAP, the users, user groups, and access control must be set up in the Yantra 7x system.

Yantra 7x also supports password expiration through LDAP. Your custom code for user authentication is interfaced with the Yantra 7x authentication mechanism. If your custom code contains ExpireInDays

with a numeric value of <X>, then a message to reset the password appears in the Yantra 7x home page. If the map contains ChangePasswordLink then the message contains a link to the location specified. Clicking on the link opens a new window with the given ChangePasswordLink.

Since the various implementations of LDAP, handle password expiration differently a sample YFSLDAPAuthenticator is modified to provide an example of one particular implementation. This is located in the <YFS_HOME>/documentation/code_examples directory.

To set properties for LDAP-based authentication:

1. Install the LDAP server (see the installation instructions from your LDAP server vendor).
2. If a JAAS-compliant provider is used, create a JAAS configuration file with the following lines:

```
LDAP
{
  // refer to the JAAS compliant service provider for the login
     module details.
  <Class Name of the Login Module as specified by the Security
  provider> required
debug=true;
};
```

3. Specify the LDAP properties described in Table 11–3.

Table 11–3 LDAP-Based Authentication Properties

Property	Description
In the yfs.properties file, uncomment and specify:	
yfs.security.authenticator	If the default implementation is used, set this property value to com.yantra.yfs.util.YFSLdapAuthenticator. **Note:** This property is deprecated.
yfs.security.ldap.factory	If the default implementation is used, this property specifies the LDAP context factory classname as in your LDAP Server configuration. Set this property value to com.sun.jndi.ldap.LdapCtxFactory.

Table 11–3 LDAP-Based Authentication Properties

Property	Description
yfs.security.ldap.url	If the default implementation is used, this property specifies the URL used to access your LDAP Server. For example, `yfs.security.ldap.url=ldap://MyServer:800.`
yfs.security.ldap.o	If the default implementation is used, this property specifies the Yantra 7x organization in your LDAP Server configuration.
yfs.security.ldap.ou	If the default implementation is used, this property specifies the Yantra 7x organizational unit in your LDAP Server configuration.
yfs.jaas.loginmodule	If using JAAS, set this property value to LDAP.
yfs.security.authenticator	If using JASS, set this property value to `com.yantra.interop.services.security`.
WebLogic startWLS startup file	
-Djava.security.auth.login.config	If you are using JAAS and WebLogic, specify the full path to your JAAS configuration file.
In the Configurator UI	
Configure organizations, organization units, and users.	All users who need to access the Yantra 7x system must be set up under the LDAP server. All Yantra 7x users must belong to the same organizational unit.

11.7 Properties for Logging

Before setting up the logging parameters, ensure that you understand the log4j utility.

For detailed information about this utility, see http://jakarta.apache.org/log4j.

To set properties for logging:

1. Copy the <YFS_HOME>/resources/log4jconfig.xml.sample file and rename it to any appropriate file name.
2. Edit the properties as described in Table 11–4, "Logging Properties".

Table 11–4 Logging Properties

Property	Description
In the log4j configuration XML File	
<priority> subelement of the <root> element	Specify the level of logging desired. Yantra recommends that you set the value of this attribute to ERROR. The following are valid values for logging levels: • ERRORDTL • ERROR • WARN • INFO • TIMER • SQLDEBUG • DEBUG • VERBOSE
<appender> subelement	At the root level, this attribute specifies the associated name and class attribute. Choose a valid log4j appender class. Each subelement can also specify the layout of the message through the <layout> subelement and can filter for levels through the <filter> subelement. Instead of hardcoding the absolute path for the log file under the appender you plan to use, Yantra recommends that customers should also use a `${LOG_DIRECTORY}` parameter in the log4jconfig.xml and invoke the JVM with a `-DLOG_DIRECTORY=<application_log_directory>/<logFileName>` option.
In the yfs.properties File	
log4j.configuration	Specify the path to the log4j XML file. For example, `log4j.configuration=/resources/MyLog4J.xml`.

Note: You cannot have non-Yantra files (such as log files) in the <YFS_HOME> directory or its subdirectories while setting up a fresh installation or upgrading from previous versions of Yantra. It is recommended to have these files in a separate directory outside <YFS_HOME>.

11.8 Properties for Integration and Agent Servers

This section describes how to set proerties for an integration or agent server. For information on running an integration or agent server, see the *Yantra 7x System Management Guide*.

To set up an integration or a agent server, specify the values as described in Table 11–5. The properties files are located in the <YFS_HOME>/resources/ directory and the script files are located in the <YFS_HOME>/bin/ directory.

Table 11–5 Integration and Agent Server Set up Properties

Property	Description
In the yfs.properties File	
yfs.context.timeout	The default value is 600 seconds. For more information, see the *Yantra 7x Performance Management Guide*.
yfs.context.reaptime	The default value is 600 seconds. For more information, see the *Yantra 7x Performance Management Guide*.
In the management.properties file	
yfs.remote.pingtime	Set this property to a value equal to the number of seconds after which the integration or agent server polls the Remote Manager for a live connection. The minimum (and default) value is 600 seconds.

Table 11–5 Integration and Agent Server Set up Properties

Property	Description
yfs.remote.reconnecttime	Set this property to a value equal to the number of seconds before the integration or agent server attempts to reconnect if the initial connection with the Remote Manager fails. The minimum (and default) value is 600 seconds.
In the Configurator UI	
Transactions and services	Determines how the integration server processes messages. For details about configuring transactions and services, see the Time-Triggered Transactions Reference in the *Yantra 7x Platform Configuration Guide*.

Note: Agent servers do not use connection pooling, instead they manage the connections internally using one connection for each thread. The required properties can be set as shown in Table 11–5. For more information on the agent server connection properties see *Yantra 7x Performance Management Guide*.

12

Configuring Yantra 7x Utilities

Yantra 7x supplies sample script files (.sh for UNIX and .cmd for Windows) that you must customize, using the directions provided in this chapter.

This chapter describes all the utilities supplied by Yantra 7x, organized by the order in which you are likely to use them. It describes generic customizations that apply to most or all utilities. Further details specific to each utility are provided throughout the rest of this guide.

This chapter provides the information required to complete step 13 of the installation checklist; as indicated by the highlighted entry:

Checklist

1. Ensure that you have necessary system requirements to install and run Yantra 7x.
2. Set up your security infrastructure.
3. Install the Yantra 7x application.
4. Install the Yantra 7x language pack (optional).
5. Install and configure your application server.
6. Install and configure your WebServer or Proxy Server.
7. Install and configure your database software.
8. Size your database.
9. Install the print server.
10. Install the weighing scale software.
11. Install the Yantra 7x Mobile application.

Checklist

12. Configure the Yantra 7x properties to use with the database, agent servers, LDAP servers, logging, and so on.
13. Configure the Yantra 7x utilities for installation, runtime, migration, and production.
14. Set up the application server for use with Yantra 7x .
15. Build your Enterprise Archive (EAR).
16. Deploy the EAR to your application server as appropriate.
17. Optionally run the configuration deployment tool to migrate your configuration data.

To set up the Yantra 7x utility script files:

1. Edit the appropriate utility in the <YFS_HOME>/bin/ directory to specify the following parameters as needed. Note that based on these parameters, the scripts automatically determine the Yantra 7x-required CLASSPATH used by the JVM:
   - YFS_HOME - the directory in which Yantra 7x is installed.
   - DB_DRIVER - the fully qualified name of the JAR file containing the JDBC database driver.
   - Add any additional parameters to the CLASSPATH to include any JAR files containing your custom classes.

   Note: On UNIX, all utilities within the <YFS_HOME>/bin/ directory must have permissions set to 755.

2. Within the script of any specific utility, add values unique to it as described in this guide.

12.1 Installation Utilities

Installation utilities enable you to install Yantra 7x. These utilities are prsesent in the <YFS_HOME>/bin directory. Some of the utilities used for installing the various configurations of Yantra 7x are "loadDefaults" and "dbverify".

loadDefaults

This utility loads the standard installation database configuration, known as the "factory defaults". For detailed information, see Section 12.1.1, "Loading the Yantra 7x Database Factory Defaults" on page 95.

dbverify

This utility verifies the changes between your database setup and the entity XML files. For detailed information, see Section 12.1.2, "Verifying the Database" on page 97.

12.1.1 Loading the Yantra 7x Database Factory Defaults

When loading the database factory defaults, Yantra 7x uses the Data Migrator described in Section 12.2.1, "Data Migrator". To load the Yantra 7x database factory defaults:

1. Ensure that the path to the Java executable is in your system path.
2. Rename the `<YFS_HOME>/resources/log4jconfig.xml.sample` file to any file name appropriate for your environment.
3. Rename `<YFS_HOME>/resources/migrator.properties.sample` to `migrator.properties`.
4. Edit the `migrator.properties` file and set the following properties:

Table 12-1 Setting up the Migrator Properties

Property	Description
<YFS_HOME>	Specify as the full path to your Yantra 7x installation directory.
yfs.migrator.xml.directory	Specify as `<YFS_HOME>/database/FactorySetup/XMLS`.
yfs.migrator.log.dir	Specify the full path to the log file directory.

5. Edit the script applicable to your operating system (`<YFS_HOME>/bin/loadDefaults.cmd` on Windows) or `<YFS_HOME>/bin/loadDefaults.sh` on UNIX and LINUX) and set the following properties:

Table 12-2 Setting up the loadDefaults Properties

Property	Description
<YFS_HOME>	Specify as the full path to your Yantra 7x installation directory.
DB_DRIVER	Specify as the full path to your database driver.
CLASSPATH	Add the `yfsdbextn.jar` file to this property.

6. Load the defaults, using the script applicable to your operating system. From the command line, run the `loadDefaults.sh` on UNIX and Linux or `loadDefaults.cmd` on Windows and pass the absolute file path to the installer:

   ```
   loadDefaults.sh <YFS_HOME>/database/FactorySetup/install/installer.xml
   ```

   Note: If the factory default installation stops before it is finished, a file name "`installer.xml.restart`" is created. This file records the location where the installation was stopped, and it is used the next time the factory defaults are installed.

7. If you plan to assign ship nodes based on US region definitions, you can also load US Region Schema defaults to populate the regions tables. To do this, run the `RegionSchema-US.sql` script located in the `<YFS_HOME>/database/FactorySetup/Optional/<database_type>/RegionSchema-US` directory that is applicable to your database.

   For example, if you are using a SQL Server database, the `RegionSchema-US.sql` script can be found at `<YFS_HOME>/database/FactorySetup/Optional/sqlserver/RegionSchema-US` directory.

   See the *Yantra 7x Platform Configuration Guide* for information about configuring organizations and their addresses as well as how to configure region schemas. See the *Yantra 7x Distributed Order Management Configuration Guide* for information about specifying regions when configuring sourcing rules.

8. If you plan to assign ship nodes based on a specific US geography location, you can also load US Zip Code defaults to populate the zip code tables. To do this, run the `US_ZipcodeLocation.sql` script

located in the <YFS_HOME>/database/FactorySetup/Optional/<database_type>/ZipCodeLocation directory that is applicable to your database.

For example, if you are using a SQL Server database, the US_ZipCodeLocation.sql script can be found at <YFS_HOME>/database/FactorySetup/Optional/sqlserver/ZipCodeLocation directory.

See the *Yantra 7x Platform Configuration Guide* for information about configuring organizations and their addresses.

12.1.2 Verifying the Database

Yantra 7x provides a database verification and correction tool to ensure database schema integrity. To set up the Database Verification Tool:

1. Set the following properties in the dbverify.sh (or .cmd on Windows) script located in the <YFS_HOME>/bin/ directory:
   - YFS_HOME - Yantra 7x installation directory
   - DB_DRIVER - Database driver JAR file
   - URL - Database JDBC URL, refer to the yfs.properties.sample for the yfs.dblogin.jdbcurl property.
   - JAVA_HOME - Your JDK installation directory.
2. From the command line, run the dbverify.sh script and pass the userID and password parameters as follows:

   ```
   dbverify.sh <userId> <password>
   ```

3. The differences between the entity XMLs and the database are generated in the form of SQL scripts, which can be run against the database to rectify the differences. The following scripts are generated:
   - Yantra_Sequence.sql - This script creates all of the additional sequences that need to be created.

> **Note:** If you are using a SQL Server database, `Yantra_Sequence.sql` is not created when you run the dbverify command.

- `Yantra_TableChanges.sql` - This script contains all the table column differences that need to be applied on the database schema. Modify this file to reference your tablespaces.
- `Yantra_TableDrops.sql` - This script removes any extra tables in the database.
- `Yantra_IndexAdds.sql` - This script adds all of the indexes that need to be created in the database. Modify this file to reference your tablespaces.
- `Yantra_IndexDrops.sql` - This script removes any extra indexes in the database.

> **Note:** The `*Drops.sql` scripts indicate extra objects in the database. These extra objects could be custom objects or objects that are dropped as the result of a schema change or an upgrade. Please look through this script carefully.

4. Run the applicable scripts to apply the changes to the database.

12.2 Upgrade Utilities

Upgrade utilities enable you to upgrade Yantra 7x. The upgrade-related utilities are described in depth in the *Yantra 7x Upgrade Guide*s.

MigrationValidator

For more information see the *Yantra 7x Upgrade Guide*.

migrator

The Data Migrator migrates your data. It has a properties file and takes in an XML file as a option.

12.2.1 Data Migrator

Yantra 7x provides a Data Migrator to ease the aspects of application configuration when loading factory defaults.

The Data Migrator reads in a task definition file which is passed to it as a parameter. These files are located in the <YFS_HOME>/Migration subfolders.

12.2.1.1 Data Migrator Task Definition Files

The Yantra 7x installation provides a task definition file that is run by the Data Migrator.

A typical task definition has the following structure:

```
<Task Class="" When="">
<TaskInfo Completed="N">
….Task specific information…
</TaskInfo>
<ChildTasks>
<Task/>
</ChildTasks>
</Task>
```

The important elements and attributes in the task definition file are described in the following sections for informational purposes only. Do not modify the task definition file.

Task Element

The <Task> element identifies one task to be processed by the Data Migrator. The attributes of the <Task> element are:

- Class - Attribute for specifying the name of the Java Class responsible to process this task. While processing this task, the Data Migrator invokes a pre-defined method in this Class and passes it the TaskInfo element as a parameter to the method. This element can be used by the Class to store and retrieve task-specific information. It is not possible to write your own implementation for this class.
- When - Attribute that sequences a group of related tasks. Valid values are:
  - First - Value that specifies which task to invoke first, before processing any related tasks (defined in the ChildTasks element).

If a task defined as First fails, the related tasks defined under ChildTasks are not processed.

- AfterChildren - Value that specifies a task is to be processed only *after* all tasks defined under ChildTasks have been successfully processed.

TaskInfo Element

The <TaskInfo> element contains task specific information and is dependent on the implementation of the class responsible for the task.

For example, Yantra 7x uses a class called the `XMLMigrator` to load and configure data into tables from XML files. This class stores a list of Entities that it needs to load into the database under the TaskInfo object. This class is used extensively in the task definition files shipped with the software.

ChildTasks Element

The <ChildTasks> element contains a list of tasks that need to be sequenced (either before or after) with respect to the containing task.

12.2.1.2 Data Migrator Restart File

While processing tasks defined in the input file, the Data Migrator periodically writes status information to a restart file in the same directory as the input task definition file. This file enables the Data Migrator to recover from incomplete runs without the need to process all defined tasks again.

If one or more tasks fail, or if the Data Migrator terminates abnormally, this file is used to complete the remaining tasks. In most cases, this file is used just for performance enhancement and individual tasks can be rerun.

It is strongly recommended that you preserve restart files across runs of the Data Migrator.

12.2.1.3 Data Migrator Log Files

Data Migrator data-related errors or warnings are logged in the error file configured in the `log4jconfig.xml` file. These errors usually result from business validation failures or contain instructions about data that the

Data Migrator could not automatically process and requires manual correction. All other errors are output to the Yantra 7x log files.

12.3 Development Utilities

Development utilities enable you to customize Yantra 7x to suit your business needs. They are for use while running Yantra 7x in development mode.

Configuration Deployment Tool

The Configuration Deployment Tool enables you to migrate configuration data from your development environment to your production environment. The tool is designed to migrate data modified as part of normal day-to-day operations.

Note that the Configuration Deployment Tool can be used to deploy configuration data that is the result of an upgrade, but it should ***not*** be used to perform the data upgrade itself.

For more information about the configuration deployment tool, see Chapter 14, "Deploying Configuration Data" on page 93.

12.4 Runtime Utilities

These utilities start processes that run in the background. The setup of these utilities is described in detail in Chapter 11, "Configuring Yantra 7x Properties".

Integration Server

An integration Server is a process that manages asynchronous services, such as messages to and from external systems. You can run the integration server using the `<YFS_HOME>\bin\startIntegrationServer` script.

The Yantra 7x integration Server allows Yantra 7x to collaborate with different systems, organizations, and businesses all through a standard, uniform interface to all systems. The Yantra integration Server runs in its own Java Virtual Machine (JVM) environment, separate from your application server.

Agent Server

The agent server utility starts processes responsible for processing transactions generated by the time-triggered transactions (agents). You can start multiple instances of an agent server using the the <YFS_HOME>/bin/agentserver.sh <server_name> script as many times as needed.

Agent Trigger

The trigger agent utility is used for scheduling time-triggered transactions.

You can override the agent criteria attributes only in the Real-time Availability Monitor. The command for triggering the Real-time Availability Monitor with override abilities is:

```
triggeragent.sh <criteriaID> -<AgentCriteriaAttribute>
<OverriddenValue>
```

To enable this override, you should pass the AgentCriteriaAttribute and OverriddenValue as additional parameters to the java class in the triggeragent.cmd (or .sh) file as follows:

```
java com.yantra.ycp.agent.server.YCPAgentTrigger -criteria %*
```

Therefore, when you invoke:

```
triggerAgent CustomCriteria -MyOverriddenParam DynamicValue
```

all the values are passed to the java class.

However, do not modify the parameters passed to the java class in the default triggeragent.cmd (or .sh) file. Make these changes in the file which you have copied and renamed from the triggeragent.cmd (or .sh). Also, the agent criteria XML code must have the flag AllowedOverriddenCriteria set to Y.

12.4.1 Setting Up the Classpath for the Runtime utilities:

The CLASSPATH for the startIntegrationServer, agentServer and triggerAgent scripts is:

- ***Must*** include: log4j-1.2.11.jar, *be.jar, activation.jar, bsf.jar, bsfengines.jar, mail.jar, yantrashared.jar, xalan.jar, xercesImpl.jar, xml-apis.jar, comm.jar, commons-collections-3.1.jar, commons-pool-1.2.jar,

yantrautil.jar, NetComponents-1.3.8.jar, and the driver jar file applicable to your database.

- If you are using WebLogic or WebSphere MQ, include their relevant JAR files and patches.
  - If you are using WebLogic with WebSphere MQ JMS, include: weblogic.jar, com.ibm.mq.jar, com.ibm.mqjms.jar, connector.jar, jms.jar, jta.jar, fscontext.jar, providerutil.jar
  - If you are using WebSphere:
    - WebSphere MQ to include $(MQ_HOME)/lib/com.ibm.mqbind.jar
    - WebSphere MQ and the *IIOP provider* URL, include: namingclient.jar, runtime.jar, com.ibm.mq.jar, com.ibm.mqjms.jar, connector.jar, jms.jar.
    - WebSphere MQ and the file provider URL, include: naming.jar, runtime.jar, messaging.jar, fscontext.jar, providerutil.jar, com.ibm.mq.jar, com.ibm.mqjms.jar, connector.jar, jms.jar, jta.jar
    - WebSphere classpath to include: ${WAS_HOME}/lib/naming.jar; ${WAS_HOME}/lib/ras.jar; ${WAS_HOME}/lib/wsexception.jar; ${WAS_HOME}/lib/bootstrap.jar; ${WAS_HOME}/lib/emf.jar; ${WAS_HOME}/lib/ecutils.jar; ${WAS_HOME}/lib/iwsorb.jar; ${WAS_HOME}/lib/namingclient.jar; ${WAS_HOME}/lib/runtime.jar; ${WAS_HOME}/lib/idl.jar; ${WAS_HOME}/lib/ffdc.jar; ${WAS_HOME}/lib/utils.jar; ${WAS_HOME}/properties; ${WAS_HOME}/lib/messaging.jar; $(WAS_HOME)/lib/j2ee.jar
- Must ***not*** include: *ui.jar.
- If you have developed custom Java classes (user exits, event handlers, and so forth), archive them to a JAR file in the <YFS_HOME>/extn/ directory and specify this JAR file.

- If you plan to extend your database, the `yfsdbextn.jar` should be at the beginning of your CLASSPATH to facilitate the purge agent to purge the custom and hang-off records.
- If the integration or agent server is running on IBM JDK1.4.* prepend the `xercesImpl.jar, xml-apis.jar,` and `xalan.jar` files to your BOOTCLASSPATH. For example on UNIX, the CLASSPATH is prepended as follows:

```
BOOTCLASSPATH="-Xbootclasspath/p:${YFS_HOME}$/lib/xercesImpl.jar:${YFS_HOME}$/lib/xml-apis.jar:${YFS_HOME}$/lib/xalan.jar"
```

12.5 Production Utilities

Production utilities enable you to monitor Yantra 7x after it has been deployed.

Get Server Details

The getServerDetails utility supplies details about all of the servers configured in your Yantra 7x installation.

12.6 Copying a WMS Node Configuration to a New Node

Yantra 7x enables you to copy the configuration of an existing WMS node to a new WMS node, along with all the participating enterprises of the existing node. This feature also allows the deletion of an existing node, provided it is the current node.

The Rapid Deployment Tool (RDT) feature also enables the onboarding of an enterprise to a node (creating a new participation), or the offboarding of an enterprise from a node (removing an existing participation).

These RDT features may be accessed through the application rules side panel of Yantra 7x Warehouse Management System in the Yantra 7x Configurator.

For more information about copying a WMS Node Configuration to a new Node, or Onboarding and Offboarding an Enterprise, see the *Yantra 7x Warehouse Management System Configuration Guide*.

13

Deploying Yantra 7x

After configuring Yantra 7x according to your business needs, you must deploy it into production as required by your application server. This chapter describes how to deploy Yantra 7x on WebLogic and WebSphere.

This chapter provides the information required to complete step 14, 15, and 16 of the installation checklist; as indicated by the highlighted entry:

Checklist

1. Ensure that you have necessary system requirements to install and run Yantra 7x.
2. Set up your security infrastructure.
3. Install the Yantra 7x application.
4. Install the Yantra 7x language pack (optional).
5. Install and configure your application server.
6. Install and configure your WebServer or Proxy Server.
7. Install and configure your database software.
8. Size your database.
9. Install the print server.
10. Install the weighing scale software.
11. Install the Yantra 7x Mobile application.
12. Configure the Yantra 7x properties to use with the database, agent servers, LDAP servers, logging, and so on.
13. Configure the Yantra 7x utilities for installation, runtime, migration, and production.
14. Set up the application server for use with Yantra 7x .

Checklist
15. Build your Enterprise Archive (EAR).
16. Deploy the EAR to your application server as appropriate.
17. Optionally run the configuration deployment tool to migrate your configuration data.

If you need to deploy Yantra 7x in a development environment using exploded (non-ear) mode, see the *Yantra 7x Customization Guide*.

Before deployment, be sure that you have applied all concepts that pertain to your environment and have completed the Performance Recommendations Checklist as described in the *Yantra 7x Performance Management Guide*.

Tip: To enable faster loading of a JSP page, pre-compile your JSP files. For information on how to do this, see the JSP Pre-compilation section of the *Yantra 7x Performance Management Guide*.

Note: If you are planning on installing any of the Yantra Packaged Composite Applications (PCA), you may want to consider delaying the building of your Enterprise Archive (EAR) until all of your PCAs are installed. Building the EAR now and for each PCA installation will not cause harm, but you will save time if you build your EAR only once after all PCAs are installed.

13.1 Yantra 7x Enterprise Archive Package

When deploying Yantra 7x on WebLogic or WebSphere, the `yantra.ear` file consists of the following files:

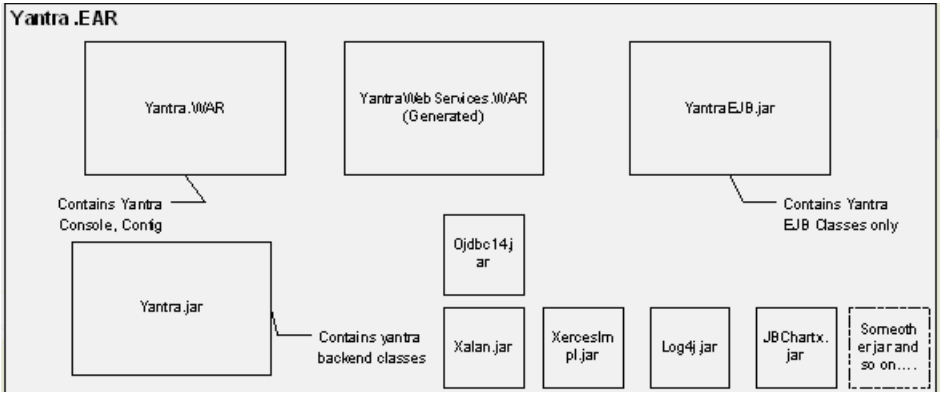
- `yantra.jar` - Application module that contains all of the Yantra 7x Java classes that encapsulate the core business logic.

- `yantra.war` - Web module that contains all of the Yantra 7x JSPs and other web application components.
- `yantrawebservices.war` - Web module that contains all of the Yantra 7x webservices interface classes.
- `yantraejb.jar` - The EJB module that contains all Yantra 7x EJBs.

Each of the third-party JAR files are left as is and in the manifest of the `yantra.jar` each file is indicated as a dependency. For example, log4j files are represented separately as `log4j-1.2.11.jar` with a dependency in `yantra.jar`.

A schematic representation of the Yantra 7x EAR on WebLogic and WebSphere is illustrated in Figure 13–1, "EAR Package in Yantra 7x":

Figure 13–1 EAR Package in Yantra 7x



The Yantra 7x online help documentation is built as a separate EAR file called `Yantradocs.ear`.

- For Weblogic, use the following command:

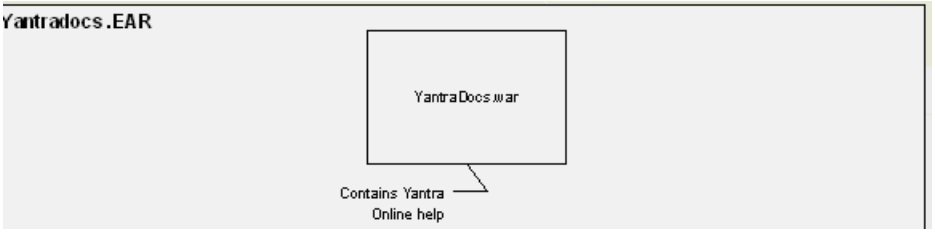
  ```
  ant -f bin/buildWLS.xml create-docear
  ```

- For WebShpere, use the following command:

  ```
  ant -f bin/buildWS.xml create-docear
  ```

A schematic representation is illustrated in Figure 13–2, "Yantra 7x Documentation Package":

Figure 13–2 Yantra 7x Documentation Package



If you want to make use of the documentation and the online help files associated with Yantra 7x, the `Yantradocs.ear` should be deployed in all of the same locations where the `yantra.ear` is deployed.

13.1.1 Build Targets

The build files for deploying Yantra 7x on WebLogic and WebSphere are `buildWLS.xml` and `buildWS.xml` respectively. These files can be accessed from the `<YFS_HOME>/bin` directory. The EAR can be created in the `<YFS_HOME>` directory.

- For Weblogic, use the following command:

  ```
  ant -f bin/buildWLS.xml create-docear
  ```

- For WebShpere, use the following command:

  ```
  ant -f bin/buildWS.xml create-docear
  ```

To execute the `buildXX.xml`, the `<YFS_HOME>/bin/build.properties` file should be configured with the appropriate application server's lib directory.

Some of the main targets for the build file are represented in Table 13–1. You can view the list of targets using the following command:

```
ant -f bin/buildXX.xml -projecthelp
```

Table 13–1 Some Targets for Creating an EAR on WebLogic and WebSphere

Main Target	Description
create-ear	Creates the Yantra 7x Enterprise Application Archive (EAR) file and the Yantra 7x documentation EAR. This is the default target.
create-docear	Creates Yantra 7x documentation EAR.
create-ejb	Creates the Yantra 7x Enterprise JavaBeans (EJB) jar file.
create-war	Creates the Yantra 7x Web Application Archive (WAR) file.
db-extn	Creates database extensions.
webservice-extn	Creates the Yantra 7x webservices extensibility.
servicegen	Instruments the EAR for expsoing Yantra 7x EJB as webservices.
update-ear	Updates the Yantra 7x EAR file.
update-docear	Updates the Yantra 7x documentation EAR file.
update-ejb	Updates the Yantra 7x EJB.
update-war	Updates the WAR file.

13.2 Deploying Custom Classes

When deploying Yantra 7x on either WebLogic or WebSphere, if you have developed custom Java classes (user exits, event handlers, and so forth) you need to deploy them in order for them to be available to Yantra 7x at runtime.

To ensure that your custom classes get invoked:

1. Create a JAR file with all your custom classes.
2. Place this JAR file in the `<YFS_HOME>/extn/` directory.
3. Include this JAR file as part of the CLASSPATH environment variable in the following scripts:
   - startIntegrationServer.sh
   - agentserver.sh

These classes are automatically included in the yantra.ear built for your application server.

13.3 Support for Mixed (Secure and Unsecure) Protocols in the Yantra 7x Application Consoles

Yantra 7x provides an optional filter that can be configured to enforce secure (https) access to certain pages and resources, such as the login page, order detail screens, or almost any other screen in the console. The only limitation is while using JavaScript and this limitation is detailed in Section 13.3.1, "Configuring and Enabling the Filter". This filter can specify patterns for URLs that need to be secured through the https protocol.

13.3.1 Configuring and Enabling the Filter

To configure the filter, copy the web.xml file to your YFS_HOME/extn directory and enable the filter by specifying the configuration shown in Example 13–1, "Enabling the Filter in web.xml File". The default web.xml file is located in the <YFS_HOME>/descriptors/weblogic/WAR/WEB-INF directory for WebLogic installations, and the <YFS_HOME>/descriptors/websphere/WAR/WEB-INF/ directory for WebSphere installations.

Example 13–1 Enabling the Filter in web.xml File

```
<filter>
   <filter-name>YantraSSLFilter</filter-name>
   <display-name>Yantra SSL Filter</display-name>
   <description>This filter will secure certain
   pages</description>
   <filter-class>com.yantra.yfc.ui.backend.filters.YFCSSLFilte
   r</filter-class>
   <init-param>
      <param-name>HTTPListenPort</param-name>
      <param-value>7777</param-value>
   </init-param>
   <init-param>
      <param-name>HTTPSListenPort</param-name>
      <param-value>7778</param-value>
   </init-param>
```

```
      <init-param>
         <param-name>SecuredURLPatterns</param-name>
         <param-value>login.jsp;personinfo.detail</param-value>
      </init-param>
      <init-param>
         <param-name>UnsecuredURLPatterns</param-name>
         <param-value>home.detail;order.detail;.search</param-val
         ue>
      </init-param>
      <init-param>
         <param-name>UnchangedQueryStringPatterns</param-name>
         <param-value>SaveAction=Y;ID=YADD</param-value>
      </init-param>
      <init-param>
         <param-name>SecuredQueryStringPatterns</param-name>
         <param-value>ID=YOMD140;ID=YOMD141;ID=YOMD143;ID=YOMD146
         ;ID=YOMD150;ID=YPMD010</param-value>
      </init-param>
      <init-param>
         <param-name>UnsecuredQueryStringPatterns</param-name>
         <param-value></param-value>
      </init-param>
   </filter>
   <filter-mapping>
      <filter-name>YantraSSLFilter</filter-name>
      <url-pattern>/*</url-pattern>
   </filter-mapping>
```

For details about the filter element and the `web.xml` file, refer to the Servlet 2.3 specification from Sun Microsystems. The filter should be mapped to intercept all requests to resources in the Yantra 7x application. Therefore, the standard filter-mapping element should be configured with a /* URL pattern.

The filter takes the following initialization parameters:

Table 13–2 Filter Initialization Parameters

Parameters	Description
HTTPListenPort	If your application is available to users on the standard HTTP port (80), you do not need to specify this parameter. If your web application is available to users on a non-standard port, specify the port being used as this parameter value.
HTTPSListenPort	If your application is available to users on the standard HTTPS port (443), you do not need to specify this parameter. If your web application is available to users on a non-standard port, specify the port being used as this parameter value. **Note:** If your deployment uses a reverse proxy for load balancing or failover in the application server cluster, these listen ports are the same as the proxy's ports.
SecuredURLPatterns	A semicolon-separated list of patterns that, if matched, would force the use of the HTTPS protocol for accessing the requested resource. This parameter does not accept wild cards or regular expressions like standard servlet URL patterns supported by application servers. The usage of this parameter is Yantra 7x-specific. To evaluate if a URL matches any of the specified parameters, Yantra 7x automatically treats each pattern as a regular expression with wild card characters on either side. For example, a pattern specified as "login" is implicitly evaluated as "*login*". Therefore, /yantra/console/login.jsp matches this pattern as does /yantra/console/extn/myloginpage.jsp. However, if the pattern is specified as "login.jsp", /yantra/console/login.jsp is treated as a match, while /yantra/console/extn/myloginpage.jsp is not.
UnsecuredURLPatterns	A semicolon-separated list of patterns that, if matched, would force the use of the HTTP protocol for accessing the requested resource.
SecuredQueryStringPatterns and UnsecuredQueryStringPatterns	A semicolon-separated list of patterns that match against the query string of the requested resource as opposed to the base URL. The pop-up screens use the CurrentDetailViewID and ActionID in their query strings so they can match specific view IDs. The secured or unsecured query string specifies the IDs that need to use the HTTPS and HTTP protocols respectively. In these cases, you can switch between HTTP & HTTPS.
UnchangedQueryStringPatterns	A semicolon-separated list of patterns that match against the query string of the requested resource. This parameter is used in some cases where you specifically do not want the switch from HTTP to HTTPS to take place. For example, when saving a form, the save should take place over the same channel as the screen itself. If a query string matches one of the specified patterns, the mode does not change from HTTP to HTTPS or vice-versa.

Note: If JavaScript is used to communicate between an HTTP and an HTTPS screen, a runtime error is raised, because the HTTP to HTTPS conversion is not supported along these boundaries.

For example, the address popup cannot be made secure. However, a custom secure popup may still work.

13.3.2 Mechanics of the filter

The filter intercepts requests for Yantra 7x Console resources and validates that the resource is being requested in a secure manner if it is so configured. Additionally, if a resource is configured as unsecured, the filter ensures that the resource is indeed being accessed through HTTP. If the resource is being requested using a protocol other than that determined by the filter, the filter redirects the request to the same resource using the desired protocol. Therefore, by the configuration depicted in the XML Example 13–1, "Enabling the Filter in web.xml File", if the `login.jsp` page is requested through HTTP, the filter would automatically redirect the browser to a secure version of the page using HTTPS on the `HTTPSListenPort`. After the secure logon is completed successfully, the browser requests the home page using the HTTPS protocol. When the filter gets that request, it determines that the home page is configured to be an unsecured resource and redirects the browser to an unsecured version of the page using the specified `HTTPListenPort`.

This filter is only useful for mixed protocol environments where only a few resources need to be secured. It is not required for deployments where there is no need to mix protocols.

Note: This filter has been tested for the standard Apache/WebLogic and Apache/WebSphere proxies supported by Yantra 7x. It also works for deployments that do not utilize a reverse proxy. While it is expected to handle deployments with most proxy servers, other proxy servers and more complex deployment models with multiple access paths and firewalls may hide actual request URLs from the filter. Therefore, Yantra recommends that you certify this filter for use in your deployment before using it.

13.3.3 Securing Login Information When Using Mixed Protocols

The Yantra 7x Application Consoles login information is protected when using mixed protocols. However, the Yantra 7x Configurator and Yantra 7x System Management passwords are not secure. Therefore, you should configure these user interfaces to suppress the password prompt.

In order to suppress the password prompt set the `yfs.config.password.noprompt` property in the `<YFS_HOME>/resources/yfs.properties` file to `Y`. If you then try to use the Configurator and System Management consoles after a session time-out, you are asked to log back in through the Yantra 7x Application Consoles

This property is set to `yfs.config.password.noprompt=N` as a default, or if not specified.

13.4 Setting Up Scripts for Creating Yantra 7x EAR

You need to set up certain scripts and classpath in your application server to create the EAR file for Yantra 7x.

- For WebLogic-specifc settings see:
  - "Setting Up the WebLogic Application Server",

  - "Configuring WebLogic for Yantra 7x" and
  - "Setting up WebLogic to Use WebServices".
- For WebSphere-specific settings see:
  - "Preparing WebSphere for Yantra 7x",
  - "Configuring WebSphere JVM Settings",
  - "Configuring a WebSphere Virtual Host for Yantra 7x", and
  - "Setting Up WebSphere to Display Barcodes and Graphs".

13.5 Creating Yantra 7x EAR

In order to create the Yantra 7x EAR see the following sections for application server-specific information:

- For creating the EAR on WebLogic, see "Creating and Deploying the Enterprise Archive on WebLogic"
- For creating the EAR on WebSphere, see "Creating the Enterprise Archive on WebSphere" and "Configuring the Enterprise Archive on WebSphere".

13.6 Deploying Yantra 7x EAR

The following sections provide indepth information specifc to an application server for deploying the Yantra 7x application:

- For deploying on WebLogic, see "Creating and Deploying the Enterprise Archive on WebLogic".
- For deploying on WebSphere, see "Deploying the Enterprise Archive in WebSphere".

13.7 Deploying Yantra 7x on WebLogic

You can pre-compile the JSPs, using the WebLogic utility `weblogic.appc`. To pre-compile the Yantra 7x JSPs, the Java CLASSPATH of the `weblogic.appc` utility must be the same as the CLASSPATH you used to start the WebLogic server for Yantra 7x.

13.7.1 Setting Up the WebLogic Application Server

If you are using HP-UX 11.22, verify that your kernel parameters are set according to BEA's recommendations before you set up the WebLogic Application Server. For these recommendations, go to http://e-docs.bea.com/platform/suppconfigs/configs81/hpux11_risc/81sp5.html.

To set up the WebLogic script file:

1. Edit the `startWebLogic.sh` file supplied by BEA as follows:

Table 13–3 WebLogic Script File Properties

Property	Description
YFS_HOME	Specify the path to your Yantra 7x installation directory.

Table 13–3 WebLogic Script File Properties

Property	Description
CLASSPATH	Modify this environment variable to include: • <YFS_HOME>/lib/bsf.jar • <YFS_HOME>/lib/bsfengines.jar • <YFS_HOME>/lib/ojdbc14.jar (for Oracle) or <YFS_HOME>/lib/db2jcc.jar and db2jcc_license_cu.jar (for DB2) or <YFS_HOME>/lib/Opta.jar (for SQL Server) • <YFS_HOME>/lib/xercesImpl.jar • <YFS_HOME>/lib/xml-apis.jar • <YFS_HOME>/lib/xalan.jar • <YFS_HOME>/lib/comm.jar • <YFS_HOME>/lib/commons-collections-3.1.jar • <YFS_HOME>/lib/commons-pool-1.2.jar • <YFS_HOME>/lib/yfcremote.jar • <YFS_HOME>/lib/log4j-1.2.11.jar • <YFS_HOME>/lib/JBChartX.jar • <WEBLOGIC_HOME>/weblogic81/server/lib/weblogic.jar **Note:** The yfcremote.jar should exist only in the CLASSPATH of the admin server, and not for the managed server. In instances where the admin server and the managed server are the same, the yfcremote.jar must exist in the CLASSPATH.
JAVA_OPTIONS	Depending on your environment, specify as follows: • For AIX, set this value to -Xms512m -Xmx512m • For JRockit, set this value to -Xms512m -Xmx512m • For HP, set as this value to -XX:MaxPermSize=128m -Xms512m -Xmx512m • For Sun, set this value to -XX:MaxPermSize=128m -Xms512m -Xmx512m

Table 13–3 WebLogic Script File Properties

Property	Description
-Dfile.encoding	To ensure that all Yantra 7x UI screens display UTF-8 characters, specify as follows for java commands: -Dfile.encoding=UTF-8 **Note:** This is applicable to all Yantra 7x java start-up scripts.

2. If you are using an HTTPS transport, download the Secure Socket Extension (JSSE) 1.0.3 package from http://java.sun.com and add the following files to the `<JAVA_HOME>/jre/lib/extn/` directory:
   - `jnet.jar`
   - `jcert.jar`
   - `jsse.jar`

13.7.1.1 Preparing Your WebLogic Setup for JNDI Cleanup

The JNDI registry records an entry for each WebLogic application server and each Yantra 7x server that starts up. These entries enable Yantra 7x to manage servers and to broadcast cached data updates to them. When a Yantra 7x server is shut down normally, the corresponding entry in the JNDI registry is removed.

When a Yantra 7x server ends abnormally (or whenever an application server ends) the corresponding entry remains in the JNDI registry, even though it no longer points to a valid running server. These pointers to servers that are no longer running are known as "stale entries." Stale entries may cause significant slowdown when managing servers through the System Administration Console and when broadcasting cache updates of configuration changes from the Yantra 7x Configurator.

To eliminate stale entries from the JNDI tree, Yantra 7x automatically tries to remove them during the initialization phase of any server start up. On WebLogic, this process can remove active entries as well as stale entries. This behavior may result in conditions ranging from benign (such as an inability to see a server in the System Administration Console) to potentially serious data integrity issues resulting from failed cache updates.

Yantra 7x supplies a script that enables you to maintain an accurate JNDI registry, using the following procedure. This script is extremely lightweight and does not require significant resources or separate sizing estimates.

To maintain accurate entries in the JNDI registry:

1. Edit the <YFS_HOME>/resources/management.properties file and set the jndi.nocleanup property to true. For example, jndi.nocleanup=true.
2. Ensure that the CLASSPATH environment variable includes the weblogic.jar file.
3. Schedule and run the <YFS_HOME>/bin/jndicleanup.sh as needed. Yantra recommends running this script every 30 minutes in your production environment.

13.7.2 Configuring WebLogic for Yantra 7x

You must configure WebLogic to run properly with Yantra 7x.

To configure WebLogic:

1. From the WebLogic Console menu, choose Services > XML.
2. Choose "Configure a New XML Registry".

> **Note:** You do ***not*** need to set an XML registry parameter for UTF-8. This is predefined.

3. Change the Document builder factory, SAX parser factory, and Transformer factory to start with "org" instead of "weblogic".
4. Target and deploy this XML registry to the WebLogic application server or cluster you want to use.

13.7.2.1 Disabling Instrumented Stack Traces in WebLogic

You can eliminate additional stack traces resulting from an error on an API call in EJB mode.

To eliminate stack traces:

1. From the WebLogic System Administration Console, select each server on which the Yantra 7x application is deployed.
2. Select Logging.
3. Uncheck the checkbox for Instrument Stack Traces and choose Apply.

13.7.2.2 Setting up WebLogic to Display Barcodes and Graphs

Yantra 7x uses the X-windows functionality to display barcodes and dynamic graphical images (such as inventory supply & demand graphs) in a UNIX environment.

The following configuration is required to enable the X-windows environment in UNIX systems for WebLogic application server:

1. If your UNIX server is also an X Windows client, edit the `startWebLogic.sh` script, and set the DISPLAY environment variable as follows:

   ```
   export DISPLAY=<IP_address_of_XWindows_server>:0.0.
   ```

2. If you are using UNIX, run the `xhost +` command to remove access control for your X Windows server.

   You can run X-server on the same server in which you run Yantra 7x. However, you need to be logged to the server console.

13.7.3 Setting up WebLogic to Use WebServices

You can have Yantra 7x run services as webservices on WebLogic and group APIs and Service Definitions into separate WAR files for load balancing. Each WAR file can be assigned to a different WebLogic server to share the load between multiple servers. This does not, however, provide fault tolerance; if a server is down, the webservices served by that server are no longer available. A single API or Service may be included in multiple WAR files.

To have Yantra 7x run services as webservices on WebLogic:

1. Edit the `<YFS_HOME>/webservices/namedwebservices.xml` file to remove any Yantra 7x APIs that you do not want exposed as named webservices and include the Services you want to expose as named

webservices. All Yantra 7x APIs are automatically available as webservices by default.

If you plan to have Yantra 7x run services as webservices on WebLogic, you need to copy the <YFS_HOME>/webservices/namedwebservices.xml.sample file to <YFS_HOME>/webservices/namedwebservices.xml and edit it to include the services you want to expose as named webservices.

In this file you can also name specific Yantra APIs that you want to expose as webservices. To expose only some APIs as webservices, set the ExposeAllAPIs attribute value to "N" and specify each API you want to expose in a Api/Name attribute. If you want to expose all Yantra APIs as webservices, you can set the ExposeAllAPIs attribute value to "Y". If the ExposeAllAPIs attribute is set to "Y", all <Api> node attributes are ignored.

This file is included by the buildWLS.xml file and ultimately included in your Yantra 7x Enterprise Archive (EAR) file. Table 13–4, "WebServices Properties on WebLogic" describes the properties you can set in the namedwebservices.xml file.

Table 13–4 WebServices Properties on WebLogic

Property	Description
ServiceName	The name of the service that you configured using the Yantra 7x Service Builder.
ExposedName	The name that is used in the Web Services Description Language (WSDL) file. This is the name that is used to call the webservice programmatically. When specifying a service name for ExposedName, choose a literal that does not match any of the standard Yantra 7x API names.

2. Generate the ycpwsbe.jar file by executing:

   ```
   ant -buildfile bin/buildWLS.xml webservice-extn
   ```

   The ycpwsbe.jar file is located in the <YFS_HOME>/extn directory.

3. Copy the <YFS_HOME>/webservices/weblogic/WEB-INF directory to a temporary directory, such as /tmp/WEB-INF.

4. Edit the /tmp/WEB_INF/weblogic.xml file. Change the context-root value to your custom name.

This is the Uniform Resource Identifier (URI) that is entered into the browser to call the webservice.

5. Edit the /tmp/WEB-INF/web-services.xml file as follows:
   a. Change the URI attribute of the web-service element to match the context-root entered in Step 4 (above).
   b. Remove all APIs and Services from this file that are to be excluded. For example, if the resulting WAR file is to contain only order-related APIs and Services, remove all others from this file.
6. Use jar or Winzip to create a WAR file with the /WEB-INF directory in the pathname.
7. The newly created WAR file should remain in the <YFS_HOME>/extn directory.
8. Repeat Step 3 through Step 7 for each WAR file you want to create.
9. Copy the <YFS_HOME>/descriptors/weblogic/EAR/META-INF/application.xml file to the <YFS_HOME>/extn directory.
10. Edit the <YFS_HOME>/extn/application.xml file to add the following element for each WAR file you created:

```
<module>  <web> <web-uri>custom war filename from </web-uri>
<context-root>custom webservice name from </context-root> </web>
</module>
```

11. Create the yantra.ear as described in Section 13.7.4, "Creating and Deploying the Enterprise Archive on WebLogic", passing the create-ear parameter into the build.sh script.

Thus, you can have Yantra 7x run as webservices by following the above mentioned steps.

To expose Yantra 7x EJBs as Webservices:

In WebLogic, you can instrument the EAR to expose Yantra 7x EJBs as webservices using the ANT task, ServiceGenTask. You can create the EJBs as webservices using the target servicegen in the buildWLS.xml located in the <YFS_HOME>/bin directory.

> **Note:** The `buildWLS.xml` is setup to ignore the ANT version that ships with WebLogic when running the `servicegen` task and use the ANT version that is specified in Chapter 2, "System Requirements" on page 3.

However, the system classpath should be unset in the calling environment. As the properties are read-only in ANT, the classpath cannot be overwritten.

13.7.4 Creating and Deploying the Enterprise Archive on WebLogic

> **Note:** Set the number of file descriptors (`ulimit -n`) for the user creating the EAR to be greater than 8192.
>
> If you are deploying on HP set `ulimt unlimited` for the user creating the EAR to be greater than 8192.

To create a yantra.ear file:

1. Set the YFS_HOME environment variable to the Yantra 7x installation directory.

   > **Note:** If you do not define YFS_HOME, you may see the error "taskdef class com.yantra.tools.ant.ForEach cannot be found".

2. Set the JAVA_HOME environment variable to the Java installation directory.
3. Set the ANT_HOME environment variable to the `<YFS_HOME>/apache-ant-1.6.5` directory.
4. Make sure `$ANT_HOME/bin` is in the PATH environment variable.
5. Set or export the `ANT_OPTS` environment variable based on your operating system (`set` for Windows and `export` for UNIX or LINUX operating systems) and JVM vendor as follows:

   `ANT_OPTS=-Xmx512m -Xms512m -XX:MaxPermSize=128m for Sun and`

HP JVMs
ANT_OPTS=-Xmx512m -Xms512m for AIX and JRockit JVMs

6. Ensure that you run all ANT commands mentioned here from the <YFS_HOME> directory.

   If you are running Yantra 7x services as webservices on WebLogic, run ant as follows:

   ```
   ant -buildfile bin/buildWLS.xml webservice-extn
   ```

   This creates the webservices JAR file called ycpwsbe.jar in the <YFS_HOME>/extn directory.

7. If you have extended your database, run the following command from the <YFS_HOME>:

   ```
   ant -f bin/buildWLS.xml db-extn
   ```

   Note: If you are running Yantra 7x on SQL Server, then you need to run the db-extn-sqlserver target before creating the yantra.ear file. To execute the db-extn-sqlserver run the following command from <YFS_HOME>:

   ```
   ant -f bin/buildWLS.xml db-extn-sqlserver
   ```

   This creates a database extension JAR file in the <YFS_HOME>/extn directory. For more information on extending your database, see the *Yantra 7x Customization Guide*.

8. From the <YFS_HOME> directory, run the following command:

   ```
   ant -buildfile bin/buildWLS.xml create-ear
   ```

   This creates the yantra.ear file in the <YFS_HOME>/drop/ directory.

   Otherwise, if you are ***NOT*** running Yantra 7x services as webservices on WebLogic, run ant as follows:

   ```
   ant -Ddonot-create-webservice=YES -buildfile
   bin/buildWLS.xml create-ear
   ```

9. Deploy your newly created yantra.ear file as described in your application server documentation.

13.7.5 Setting Up WebLogic to Use HTTP In-Memory Session Replication

The Yantra 7x application supports HTTP in-memory session replication on the following configuration:

- BEA WebLogic on HP-UX 11i
- Apache 2.0.44 with the WebLogic plug-in as the proxy server with idempotent set to OFF

We advise testing session replication if you are using a different proxy.

> **Note:** The idempotent flag must be set to OFF. In rare cases, for example when a transaction completes and commits but was unable to post the response to the proxy server, the proxy server could retransmit the transaction. For some update transactions, this could result in duplicate update entries.

The `weblogic.xml` file should be edited to set up WebLogic for in-memory session replication as follows:

1. Copy the `<YFS_HOME>/descriptors/weblogic/WAR/WEB-INF/weblogic.xml` file to the `<YFS_HOME>/extn` directory.
2. Add the following lines to the `<YFS-HOME>/extn/weblogic.xml` file:

```
<session-descriptor>
    <session-param>
        <param-name>PersistentStoreType</param-name>
        <param-value>replicated</param-value>
    </session-param>
</session-descriptor>
```

13.8 Deploying Yantra 7x on WebSphere

In order to deploy Yantra 7x on WebSphere, you need to perform the following tasks:

- Configure options relevant to the WebSphere application server
- Create and deploy a `yantra.ear` file for WebSphere

- Clear the browser and Java Plugin cache on client computers

13.8.1 Preparing WebSphere for Yantra 7x

Before configuring WebSphere, Yantra recommends that you start the WebSphere administrative server with the following memory parameters:

- -Xms128 MB or higher
- -Xmx256 MB or higher

To avoid the warning messages regarding direct datasource lookups that occurs at run time:

1. From the WebSphere Administrative Console, expand `Troubleshooting` in the left panel and click on `Logs and Trace`.
2. Select each server that hosts Yantra 7x and choose `Change Log Detail Levels` in the `General Properties`.
3. In the `Components` panel, select the class, `com.ibm.ejs.j2c.ConnectionFactoryBuilderImpl`, and specify the log level as `severe`.

13.8.2 Configuring WebSphere JVM Settings

You need to use the WebSphere Administrative Console to specify the JVM settings. These JVM settings must be set on ***all*** servers in a cluster (if you are using a cluster).

To configure JVM setting on WebSphere:

1. From the WebSphere Administrative Console, select the application server specified for Yantra 7x.
2. For IBM servers with IBM JDK 1.4.2:
   a. Select Server Infrastructure > Java and Process Management > Process Definition > Environment Entries.
   b. Choose `New` and specify the following values and then choose OK:

Table 13–5 WebSphere Properties for JVM Settings

Name	Value	Description
PSALLOC	early	PSALLOC

Table 13–5 WebSphere Properties for JVM Settings

Name	Value	Description
NODISCLAIM	true	NODISCLAIM

3. Select Server Infrastructure > Java and Process Management > Process Definition > Java Virtual Machine.
4. In the Java Virtual Machine, edit the CLASSPATH field to include the following:
   - <YFS_HOME>/lib/xml-apis.jar
   - <YFS_HOME>/lib/xercesImpl.jar
   - <YFS_HOME>/lib/xalan.jar
   - <YFS_HOME>/lib/bsf.jar
   - <YFS_HOME>/lib/bsfengines.jar
   - <YFS_HOME>/lib/comm.jar
   - <YFS_HOME>/lib/commons-collections-3.1.jar
   - <YFS_HOME>/lib/commons-pool-1.2.jar
   - Path to your database driver

   You must set these properties on EVERY application server that you plan to use. You may choose to set these properties on one server and create other servers using this server as the template. The classpath listed in the bulleted list are contiguous and are separated by a ":". For example, <YFS_HOME>:<YFS_HOME>/lib/xml-apis.jar.
5. Under the Custom Properties section, set the JVM settings to the following values:

Table 13–6 Custom Properties for JVM Settings

Name	Value	Description
client.encoding.override	UTF-8	Enables the use of special characters.

6. Restart the application server to enable these changes to take effect.

13.8.3 Configuring a WebSphere Virtual Host for Yantra 7x

You need to use the WebSphere Administrative Console to specify the server port numbers to configure a virtual host for Yantra 7x.

To configure a virtual host in WebSphere:

1. From the WebSphere Administrative Console, select the application server specified for Yantra 7x.
2. Select Container Settings > Web Container Settings > Web Container Transport Chains.
3. Create a new transport chain. Specify any name of your choice and select the `WebContainer` template. Click Next.
4. Specify a port name. Specify `host=*` and specify a port number that corresponds to one of the port numbers configured for the desired virtual host previously created.
5. Confirm the selected details and click `Finish`.

13.8.4 Setting Up WebSphere to Display Barcodes and Graphs

Yantra 7x uses the X-windows functionality to display barcodes and dynamic graphical images (such as inventory supply & demand graphs) in a UNIX environment.

The following configuration is required to enable the X-windows environment in UNIX systems for the WebSphere application server:

1. From the WebSphere Administrative Console, go to Servers > Application Server and select the application server specified for Yantra 7x.
2. On the Configuration tab, select Java and Process Management under Server Infrastructure option.
3. Select Process Defintion.
4. On the configuration, go to Additional Properties and select Environment Entries.
5. Select New.

6. On the General Properties enter the Name as DISPLAY and the value as <IP_address_of_XWindows_server>:0.0. Do make sure that the X-Windows accept requests from this client.

7. If you are using UNIX, run the xhost+ command to remove access control for your X Windows server.

   You can run X-server on the same server in which you run Yantra 7x. However, you need to be logged to the server console.

   Restart the application server for the DISPLAY variable to take effect.

13.8.5 Creating the Enterprise Archive on WebSphere

If you plan to have Yantra 7x run services as webservices on WebSphere, you need to copy the <YFS_HOME>/webservices/namedwebservices.xml.sample file to <YFS_HOME>/webservices/namedwebservices.xml and edit it to include the services you want to expose as named webservices.

In this file you can also name specific Yantra 7x APIs that you want to expose as webservices. To expose only some APIs as webservices, set the ExposeAllAPIs attribute value to N and specify each API you want to expose in an Api/Name attribute. If you want to expose all Yantra 7x APIs as webservices, you can set the ExposeAllAPIs attribute value to Y. If the ExposeAllAPIs attribute is set to Y, all <Api> node attributes are ignored.

This file is included in the buildWS.xml file and ultimately included in the Yantra 7x Enterprise Archive (EAR) file. Therefore, you must edit the attributes of the namedwebservices.xml file, as described in Table 13–7, "WebServices Properties on WebSphere", ***before*** you create your yantra.ear file.

Table 13–7 WebServices Properties on WebSphere

Property	Description
ServiceName	The name of the service that you configured using the Yantra 7x Service Builder.
ExposedName	The name that is used in the Web Services Description Language (WSDL) file. This is the name that is used to call the webservice programmatically. When specifying a service name for ExposedName, choose a literal that does not match any of the standard Yantra 7x API names.

To expose Yantra 7x EJB's as Webservices:

In WebSphere, you can expose Yantra 7x EJBs as webservices using the ANT task servicegen provided in buildWS.xml located in the <YFS_HOME>/bin directory.

You can configure Yantra 7x as webservices by:

1. Renaming the <YFS_HOME>/webservices/namedwebservices.xml.sample to namedwebservices.xml.
2. Editing the XML file to update the exposed names to start with a lowercase letter.
3. Running the following ant targets from <YFS_HOME>:

```
ant -f bin/buildWS.xml create-ear
ant -f bin/buildWS.xml servicegen
```

However, the system classpath should be unset in the calling environment. As the properties are read-only in ANT, the classpath cannot be overwritten.

To create a yantra.ear file:

1. Set the YFS_HOME environment variable to your Yantra 7x installation directory.

> **Note:** If you do not define YFS_HOME, you may see the error "taskdef class com.yantra.tools.ant.ForEach cannot be found".

2. Set the ANT_HOME environment variable to the <YFS_HOME>/apache-ant-1.6.5 directory.
3. Make sure the ANT_HOME/bin directory is included in the PATH environment variable. For more information, see the ANT documentation.
4. Set the JAVA_HOME environment variable to <WebSphere_Home>/java where <WebSphere_Home> is the WebSphere installation directory.

5. Set or export the ANT_OPTS environment variable based on your operating system (set for Windows and export for UNIX or LINUX operating systems) and JVM vendor as follows:

   ANT_OPTS=-Xmx512m -Xms512m -XX:MaxPermSize=128m for Sun and HP JVMs
   ANT_OPTS=-Xmx512m -Xms512m for AIX and JRockit JVMs

6. Ensure that you run all ANT commands mentioned here from the <YFS_HOME> directory. If you are running Yantra 7x services as webservices on WebSphere, run ant from the <YFS_HOME> directory as follows:

   ```
   ant -buildfile bin/buildWS.xml webservice-extn
   ```

   This creates the webservices ycpwsbe.jar file in <YFS_HOME>/extn directory.

7. If you have extended your database, run the following command from the <YFS_HOME>:

   ```
   ant -f bin/buildWS.xml db-extn
   ```

   Note: If you are running Yantra 7x on SQL Server, then you need to run the db-extn-sqlserver target before creating the yantra.ear file. To execute the db-extn-sqlserver run the following command from <YFS_HOME>:

   ```
   ant -f bin/buildWS.xml db-extn-sqlserver
   ```

   This creates a database extension JAR file in the <YFS_HOME>/extn directory. For more information on extending your database see the *Yantra 7x Customization Guide*.

8. From the <YFS_HOME> directory, run the following command:

   ```
   ant -buildfile bin/buildWS.xml create-ear
   ```

   This creates a yantra.ear file in the <YFS_HOME>/drop/ directory.

13.8.6 Configuring the Enterprise Archive on WebSphere

Follow these steps only if you are planning to use database connection pooling. For example, you can configure the Websphere application for connection pooling by:

1. Configuring the datasource (non-type 4) in WebSphere with the JNDI name as `jdbc/<datasourcename>`.
2. Testing the connection pooling in WebSphere.
3. Adding the `yfs.login.datasource.name=jdbc/<datasourcename>` in the `yfs.properties` file.

13.8.6.1 Importing the EAR

To add resource references to the EAR for deployment on WebSphere:

1. In binary mode, FTP the EAR file you created earlier to the Windows or Linux machine you are using.
2. Launch RAD.
3. Open the Project Explorer.
4. In the Project Explorer view, right-click on the Enterprise Applications folder and select Import > Import EAR.
5. In the dialog box that appears use the Browse button to select the EAR file you transferred in Step 1.
6. Accept the defaults and click Finish.

13.8.6.2 Adding Resource References to an EJB Module

To add a resource reference to an EJB module in RAD:

1. Expand the EJB Projects folder in the Project Explorer view.
2. Expand your yantra_EJB project.
3. Right-click on the EJB project whose resource references you want to add by selecting Open With > Deployment Descriptor Editor.

   The EJB Deployment Descriptor editor displays.
4. Locate the References tab in the editor and click Details.
5. On the Reference tab, choose the resource reference you want to create and click Add. The Add Reference dialog box displays.

6. Select the Resource Reference and click Next.
7. Enter the details of the resource reference as follows:
   a. Name: <Name of your DataSource>
   b. Type: javax.sql.DataSource
   c. Authentication: Application
   d. Sharing scope: Shareable
8. Click Finish on the Resource Reference Details screen.
9. Enter the JNDI Name in the WebSphere Bindings tab of the Reference panel. This should match the JNDI name for the datasource you created in the WebSphere Administrative console.
10. Choose Save.

This creates a datasource resource reference for the EJBs in the Yantra 7x application.

13.8.6.3 Adding Resource References to an WAR Module

To add a resource reference to a WAR module:

1. Expand the Dynamic Web Projects folder in the Project Explorer view.
2. Expand the yantra_WEB project that you need to modify.
3. Right-click on the Web module whose resource references you want to add and select Open With > Deployment Descriptor Editor.

   The Web Deployment Descriptor editor displays.
4. Locate the References tab in the editor and click Details.
5. The Reference panel displays. Click Add. The Add Reference dialog box displays.
6. Select Resource Reference and click Next.
7. Enter the details of the resource reference as follows:
   a. Name: <Name of your Resource Reference>
   b. Type: javax.sql.DataSource
   c. Authentication: Application
   d. Sharing scope: Shareable

8. Click Finish.
9. Enter the JNDI Name in the WebSphere Bindings tab of the Reference panel. This should match the JNDI name for the datasource you created in the WebSphere Administrative console.
10. Choose Save.

This creates a resource reference to a WAR module.

13.8.6.4 Exporting the EAR for Deployment

To export the EAR for deployment:

1. Expand the Enterprise Applications folder in the Project Explorer view.
2. Right-click on the EAR file and select: Export > Export EAR file.
3. Choose Browse and provide Deployed_yantra.ear as the name for exporting this EAR file. This enables you to distinguish between the EARs with and without the resource references.
4. Check the Overwrite Existing Files Without Warning checkbox.
5. Choose Finish.
6. Once the export has completed, close the RAD.
7. In binary mode FTP the new EAR file back to the original machine.

13.8.7 Deploying the Enterprise Archive in WebSphere

1. From the WebSphere Administrative Console menu in the left pane, select Applications > Install New Application. The right pane is populated with the specifics for the EAR location.
2. Choose Local File System or Remote File System. Then click the corresponding Browse button and browse to the enterprise archive such as `yantra.ear` or `yantradoc.ear` you want to deploy. Then click Next.
3. Select the Use default virtual host name for Web Modules radio button under Virtual Host and specify your virtual host name. Accept all other defaults and click Next.
4. Check Deploy enterprise beans, and if desired, change the application name. If you are using webservices, check Deploy WebServices and click Next.

5. The Map Modules to Servers screen displays. Select the checkbox next to each desired module (at least two entries YantraEJB and yantra.war should be present). Then click the Cluster/Server in the Cluster and Server pane. Click Apply. The screen refreshes and the server field is updated with the chosen value. Click Next.
6. Specify the following JAR files in the EJB deploy classpath and choose your database version for the DB Type:
   - <YFS_HOME>/lib/xercesImpl.jar
   - <YFS_HOME>/lib/xml-apis.jar
   - <YFS_HOME>/lib/xalan.jar

   Choose Next.
7. Accept the default JNDI names for the EJB modules on the Provide JNDI Names for Beans screen. Click Next.
8. On the Map Virtual Hosts for Web Modules screen, select your web module and its correct virtual host. Choose Next.
9. The Ensure all Unprotected 2.x Methods screen displays. Click Next.
10. The Provide Options to perform the WebServices Deployment screen displays. Leave them as is and click Next.
11. Choose Finish.
12. If the installation is successful, choose the Save to Master Configuration link.
13. Select the Synchronize changes with Nodes checkbox and click Save.
14. Start your application server by selecting Servers > Application Servers in the left pane. Select the checkbox next to the desired server and click the Start button.
15. Start your application by selecting Applications > Enterprise Applications in the left pane. Select the checbox next to the application you had just installed and click the Start button.

13.8.8 Application Clients Invoking Yantra 7x EJBs

In order to make EJB calls in Yantra 7x using WebSphere you need to generate EJB stubs and skeletons. The following steps outline the method

for creating the JAR files using the ejbdeploy.sh script to generate the stubs:

1. Set the CLASSPATH to include xercesImpl.jar, xalan.jar, and xml-apis.jar as provided in the <YFS_HOME>/lib directory.
2. Invoke ejbdeploy.sh from the <WebSphere_Home>/bin directory with the following three arguments:
   a. Specify the full path to the yantra.jar file in <YFS_HOME>/drop directory.
   b. Specify the temporary directory that is used for the EJB deployment.
   c. Specify the full path to the desired output file, for example yantra_ejbstubs.jar.

Additionally set the classpath on the ejbdeploy.sh command line following the -cp argument. For example:

```
$<WebSphere_Home>/bin/ejbdeploy.sh <YFS_HOME>/drop/yantraejb.jar
<WebSphere_Home>/temp <YFS_HOME>/drop/yantra_ejbstubs.jar -cp $CLASSPATH
```

The resulting output file yantra_ejbstubs.jar, is to be used by the client applications calling the EJBs.

13.9 Setting the Client Character Display

When displaying special characters, such as for various languages, the client computer must be configured to display these characters.

In order for Unicode characters to display correctly in the Yantra 7x Application Consoles, each Windows client must be configured. To configure a client machine select Control Panel > Display > Appearance.

13.10 Clearing Browser and Java Plugin Caches

Once Yantra 7x is ready for deployment, each user must clear the browser and Java Plugin caches on their client machines before launching Yantra 7x.

To clear the browser cache:

1. From the Windows start menu, select Settings > Control Panel > Internet Options. Choose the General tab, and in the Temporary Internet Files inner panel, choose the Delete Files button. The Delete Files dialog displays.
2. Enable the Delete All Offline Content option. Then choose OK, and choose OK once more.
3. Close the Internet Properties window.

To clear the Java plugin cache:

1. From the Windows start menu, select Settings > Control Panel > Java Plugin. Choose the Cache tab.
2. Choose the Clear JAR Cache button then choose OK.
3. Close the Java Plugin Control Panel window.

13.11 Verifying Your Yantra 7x Deployment

To verify the Yantra 7x installation:

1. Restart your application server.
2. Start Internet Explorer.
3. Access `http://<hostname>:<port>/yantra/console/login.jsp`.
4. When prompted for a Login ID and Password, enter "yantra" for both. If the Yantra 7x Administrator's home page is not displayed, contact Yantra Technical Support Services at 1-877-YANTRAS or http://www.yantra.com/services-technical.htm.

13.12 Statistics Monitoring

In order to measure throughput performance, runtime statistics can be gathered. Note that this feature and the data gathered by it in the YFS_STATISTICS_DETAILS table is only for the use of Yantra personnel, as any metrics can change without notice.

In a production environment, you should leave statistics generation enabled to collect statistics data in 10 minute intervals (the default). You

should also schedule statistics purges on a regular basis (for example, every two weeks).

14

Deploying Configuration Data

During incremental configurations of Yantra 7x, changes are typically developed in a test environment and then they are rolled out into production. Migrating configuration data can be fairly cumbersome and time consuming. Yantra 7x provides a Configuration Deployment Tool that enables you to migrate configuration data. This tool ensures data integrity while reducing the system downtime to transfer data and minimizing the effort needed to ensure accuracy.

This tool is designed to migrate data that is modified as part of a normal day-to-day operation. Note that the Configuration Deployment Tool can be used to deploy configuration data that is the result of an upgrade, but it should ***not*** be used to perform the data upgrade itself.

This chapter provides the information required to complete step 17 of the installation checklist; as indicated by the highlighted entry:

Checklist

1. Ensure that you have necessary system requirements to install and run Yantra 7x.
2. Set up your security infrastructure.
3. Install the Yantra 7x application.
4. Install the Yantra 7x language pack (optional).
5. Install and configure your application server.
6. Install and configure your WebServer or Proxy Server.
7. Install and configure your database software.
8. Size your database.
9. Install the print server.

Checklist

10. Install the weighing scale software.
11. Install the Yantra 7x Mobile application.
12. Configure the Yantra 7x properties to use with the database, agent servers, LDAP servers, logging, and so on.
13. Configure the Yantra 7x utilities for installation, runtime, migration, and production.
14. Set up the application server for use with Yantra 7x .
15. Build your Enterprise Archive (EAR).
16. Deploy the EAR to your application server as appropriate.
17. Optionally run the configuration deployment tool to migrate your configuration data.

Configuration Deployment Tool Features

The Configuration Deployment Tool (CDT) can be accessed from the Yantra 7x Development and Deployment WorkBench (also known as the "WorkBench").

Note: The Configuration Deployment Tool runs only on Windows.

The CDT provides the following capabilities:

- Transfer complete and partial sets of configuration data or discrete logical portions.
- Transfer data to and from XML files or databases.
- Transform certain data-like IP addresses and port numbers that are different in two environments, depending upon network configuration.
- Generate a report of configuration differences by comparing the two systems.

14.1 Concepts Regarding Configuration Data Deployment

This section defines terms relevant to the Configuration Deployment Tool, and explains how they apply.

14.1.1 Source and Target Environments

The Configuration Deployment Tool deploys data from one Yantra 7x environment to another. The deployment could occur from development to test environments, from staging to production environments, and so forth. The environment that serves as the point of origin for the data is known as the "source" environment. The destination environment into which data is deployed is defined as the "target" environment. This deployment can be in the form of importing and exporting data to and from databases or XML files. Example 14–1, "Configuration Data XML" shows the required format of these XML files.

Example 14–1 Configuration Data XML

```
<?xml version="1.0" encoding="UTF-8" ?>
<YFS_ATP_RULESList TableName="YFS_ATP_RULES">
    <AtpRules AccumulationDays="730" AdvanceNotificationTime="0"
    AtpRule="DEFAULT" AtpRuleKey="DEFAULT" AtpRuleName="DEFAULT"
    BackwardConsumptionDays="730" ConsiderPoForAlloc="N" Createprogid=""
    Createts="" Createuserid="SYSTEM" ForwardConsumptionDays="730" Lockid="0"
    MaxInventoryHoldDays="730" Modifyprogid="SYSTEM" Modifyts=""
    Modifyuserid="SYSTEM" OrganizationCode="DEFAULT" PastDueDemandDays="730"
    PastDueSupplyDays="730" ProcessingTime="0" />
</YFS_ATP_RULESList>
```

14.1.2 Configuration Groups and Driver Entities

The entire set of Yantra 7x configuration data is broken down into logical subsets called "configuration groups" and "driver entities." Configuration groups and driver entities are predefined and cannot be changed.

During the deployment process, if you need to perform more granular inserts, updates, and deletes so that your target database matches your source, you choose these configuration groups or driver entities.

Driver Entities

Most Yantra 7x configuration data can be deployed starting with a logical entity, for example, an organization or a pipeline. These logical entities are called "driver entities". Driver entities represent the most granular level of information that can be deployed from the source to the target without loss of data integrity.

Only driver entities allow deployment at a record level. For other tables either of the following conditions apply:

- The table is completely deployed if it is not dependent on any driver entity.
- Only records corresponding to the driver entity are deployed.

Information about driver entities can be stored in multiple tables and when deploying an entity, data in all related tables is deployed together in one transaction boundary to preserve data consistency.

Configuration Groups

Logically related tables or driver entities are also grouped together into "configuration groups" that typically represent larger, significant logical data models within Yantra 7x. Examples include the Business Process Model or the Participant Model. These groups are provided for convenience and for ease of navigation on the user interface.

14.1.3 Externally Maintained Configuration Data

In your implementation of Yantra 7x, you may be required to import certain data into your target that is not part of your source Yantra 7x database. For these tables, you should not use the CDT to deploy data, because the CDT does not have access to the correct data.

Best Practices

If you must use the Configuration Deployment Tool to deploy externally maintained data, the recommended way to handle this is to import this data into the source and then use the CDT to deploy it into the target. This guarantees data integrity.

If you cannot import this data into your source database, Yantra 7x supplies features that enable you to work with external data by ensuring that the target database either ignores these tables or appends them. Use the Ignore and Append-only features ***only*** as a last resort and only after subjecting your environment to rigorous testing.

Caution: When using the Ignore or Append-only features, the CDT cannot guarantee the integrity of any external data. In order to ensure data integrity, the CDT must have complete access to the configuration data.

Ignore

In cases where data in tables is maintained externally, you can omit these tables from the deployment operation by specifying a preference for them to be ignored.

Ignoring a table or a driver entity also automatically ignores all its dependent tables. However, there are some tables that store data for multiple driver entities and are present in multiple groups. An example of this is the YFS_GRAPH_UI table that contains data for pipelines, services and statuses. Ignoring one of these tables causes the CDT to incorrectly mark the corresponding records for deletion.

Append-only

In cases where some tables are partially maintained externally, you can specify preferences to ensure that these tables are deployed in an "append-only" mode.

For append-only tables, the dependent tables are not ignored. Marking a table as append-only implies that only a few rows in the target database are maintained on the source system—other rows are externally imported. In such cases, it is ***extremely*** important that there is no overlap between the data present in the source and the external system. For example, if you maintain your shipping nodes in the source database and import store information directly into the target, you must not have any stores in the source database. This leads to unpredictable results.

14.1.4 Deploying Database Extensions

Any database extensions you have defined within the Yantra 7x database extensibility framework are automatically deployed when configured to do so, as described in Section 14.3, "Setting Up the Configuration Deployment Tool" on page 150.

14.1.5 Deploying Custom Tables

The CDT automatically deploys Yantra 7x configuration tables and extensions defined within the Yantra 7x database framework. If you have custom (non-Yantra 7x) configuration tables defined in your installation, the CDT needs to be specially configured to deploy these tables. To enable the CDT to deploy these tables, the tables need to be registered with the CDT by creating a special custom deployment XML file, called `cdt_custom.xml`. A sample of this file can be found in your `YFS_HOME/resources/ydkresources` directory. This file defines a group named "Custom Tables" and should include a list of your custom tables. The CDT automatically compares, displays and deploys changes to custom records for all tables that have one or more primary key columns.

This tool does not support custom tables as drivers or the representation of custom tables in a dependency tree structure. As a result, all custom tables can only be deployed together as part of the "Custom Tables" group. It also does not support custom tables without a primary key.

The `cdt_custom.xml` file contains the following:

```
<Group Name="Custom Tables">
    <Table Name="CUSTOM_CONFIG_TABLE_1"/>
    <Table Name="CUSTOM_CONFIG_TABLE_2"/>
</Group>
```

14.1.6 Foreign Key Checks

The CDT enforces data consistency by deploying all related tables that define an entity together in one operation. In addition, to ensure data integrity, the CDT also checks the required foreign key constraints for each table - which could potentially be defined for a table in a completely different group. Therefore, when deploying a small subset of data, it is possible that you may see error messages indicating foreign key constraint violations if the corresponding data in the independent table is not being deployed in the same operation. In this case, you should try deploying a bigger set of data. Note that foreign key constraints are **not** defined or checked for custom tables.

To provide the best performance, foreign key constraints are not checked when deploying the complete Yantra 7x configuration.

14.1.7 Data Transformations

Frequently, the development and production environments have different values for network settings such as server names and IP addresses. Some configuration data tables in Yantra 7x store host names, IP addresses, and URLs. While these are valid for your source environment, when deploying this data into the target environment, the configuration must be updated with the corresponding values applicable to the target environment. The CDT enables you to automatically transform these data elements into target-appropriate values by letting you specify transformations to be carried out on the source data *before* it is deployed into the target.

14.2 Before You Begin

Before you begin using the Configuration Deployment Tool, ensure that you have addressed the policies and system requirements described in this section.

14.2.1 System Requirements

The RAM requirements of the Configuration Deployment Tool depend on the size of your database and the distribution of your configuration data. For system requirements of Yantra 7x tools, including the Configuration Deployment Tool, see Chapter 2, "System Requirements" on page 3.

Time Estimates

The time required for the Configuration Deployment Tool to perform comparison and deployment tasks varies according to your system resources and the size and distribution of your configuration data. For example, processing time may increase when there are many records in a table that are referenced by foreign key constraints from other tables, or when there are many records in a table that serves as a driver entity.

During tests of the Configuration Deployment Tool, Yantra measured the amount of time it took to perform tasks on a Pentium class machine with 512 MB of RAM and running at 550 MHz. The Configuration Deployment Tool performed as described in Table 14–1:

Table 14–1 Configuration Deployment Tool Time Estimates

Task	Description of Databases	Time
Comparison	Source database - 110,000 records Target database - 110,000 records	7 minutes
Comparison	Source database - 110,000 records Target database - empty	4 minutes
Deployment	110,000 differences	11 minutes

Installation

The Configuration Deployment Tool is installed automatically during the Yantra 7x installation process.

Environment State

The Configuration Deployment Tool assumes that your source and target environments match exactly in the following respects:

- Release of Yantra 7x (including hot fixes)
- Release of JDBC drivers
- Release of database software
- Database structure (Yantra 7x schema objects such as tables, indexes, and sequences)

As this Configuration Deployment Tool is used by technical professionals for tasks they perform on an occasional basis, it is **not** localizable or customizable. However, you can specify configuration preferences as described in this chapter.

14.2.2 Security Strategy

The Configuration Deployment Tool makes use of the user authentication and authorization supplied by your database provider. Access control and authorization are not specified through the Configuration Deployment Tool.

Ensure that the person using the Configuration Deployment Tool has sufficient authentication privileges (select, insert, update, and delete) for both databases; full DBA privileges are not required.

14.2.3 Change Management Strategy

The Configuration Deployment Tool does **not** enforce checks to restrict configuration data modification on the source or the target schemas using other means. You must develop and enforce your own methodology.

For example, if you use the Configuration Deployment Tool to migrate data from staging to production, it is **not** expected that the configuration in production is modified by means other than this tool. In such a case those changes are overwritten the next time the CDT is run. Also, this could potentially lead to data integrity issues if the changes are performed in either the source or target while the CDT is being run.

The Configuration Deployment Tool is **not** supported for implementations where configuration data is directly modified in production using the Yantra 7x Configurator or any other means. For exceptional cases like urgent or critical fixes to configuration data in production, you must update the staging database with the same changes.

14.2.4 Rollback Strategy

To prevent application failure and downtime, implement a rigorous rollback methodology that involves creating a backup snapshot of your configuration data in production **before** you use the CDT to deploy changes. This backup can be accomplished by using the database-specific export and import utilities. Yantra 7x provides some samples for Oracle, DB2 and SQL Server database that you can customize for your own use.

14.2.5 Upgrades and Maintenance

Using the Configuration Deployment Tool should not impact the methodology for applying upgrades or hot fixes in a multi-step staging environment.

The upgrade methodology being followed should not change for environments already set up for staging before production. However, the Configuration Deployment Tool by itself does **not** provide support for all of the processes and methodologies required for supporting a multi-step application staging and deployment environment because it is only capable of deploying configuration data.

The process of applying product upgrades and patches is especially complex in an environment where the staging area must be kept synchronized with production. One way to keep these environments harmonious is to apply software patches to both systems simultaneously and reverse deploy the data upgrades. This is because application data upgrades may behave differently and produce different results based on the transactional data they encounter. If this application data upgrade is run independently on production and staging, the results may be significantly different as a result of the differences in transactional data that the upgrade program encountered. In such a case, the production snapshot should be treated as the baseline and reverse deployed into staging. This can be accomplished by configuring your production database as the source and your staging database as the target.

Example Upgrade Scenario

In an example upgrade scenario, Yantra 7x introduces a feature that recognizes various attributes for order types. For example, an Order_type "URGENT" implies that the order should be displayed in the user interface with a specific icon that enables you to distinguish it from other orders. However, in previous releases of Yantra, you may have been using the Order_type field to classify orders into other types because this field was designed for order classification.

When Yantra provides an upgrade toolkit, one component of the toolkit handles upgrades to the Order_type field.

The upgrade logic may flow as follows:

1. Read all the distinct values of the Order_type field from the YFS_ORDER table.
2. For each different Order_type in the your system, create entries in the Order_Type_Master configuration table and assign a default icon to it.

If this data upgrade is run on the staging system, it will **not** find any orders, so the Order_Type_Master table will only contain "URGENT" which is provided by default.

However, when the same data upgrade is run in production, the Order_Type_Master table will contain multiple entries, one for each type of order that is in the transaction database.

Then, when the Configuration Deployment Tool is run again, it marks all of these new records for deletion because the source is assumed to be the most up-to-date configuration. This behavior is obviously **incorrect**.

As a result, you should design upgrade kit or hot-fixes for transaction dependent configuration data as follows:

1. The upgrade kit (or hot-fix) should have one script to prepare input for upgrade of transaction-dependent configuration data (for example, prepared list of distinct order types). Then you can run this script on the production database. You can also run this script in the test database and can take the union of the two.
2. The next step in the upgrade should use this as input and upgrade the configuration data accordingly. For example, inserting into ORDER_TYPE_MASTER table.

If you have identified any changes in the configuration data, please contact Yantra Technical Support Services at 1-877-YANTRAS.

14.2.6 Externally Maintained Configuration Data

In your implementation of Yantra 7x, you may have external data that is not part of your source database, as described in Section 14.1.3, "Externally Maintained Configuration Data" on page 142. If you are required to import this data, you should devise an external method rather than using the Configuration Deployment Tool.

If you ***do*** use the Configuration Deployment Tool to deploy this data, you can configure it to omit these externally maintained tables or to only append specific data as described in Section 14.6, "Specifying the Preferences Settings" on page 158.

Note that these options may incorrectly cause the CDT to mark some records as "Abandoned" since the data for the corresponding driver entity cannot be identified. For details on working with Abandoned records, see Section 14.15, "Troubleshooting" on page 171.

14.2.7 Best Practices

When deploying data into production or other critical environments that feed into it, use the CDT to compare and deploy the ***entire*** configuration

rather than an individual configuration group or driver entity. Deploying the entire database addresses the following issues:

- Data integrity - Deploying an individual configuration group or driver entity is most likely to result in messages about data integrity violations. Understanding how to address these messages requires a significant understanding of the Yantra 7x Data Model.
- Custom tables - The CDT only deploys custom configuration tables when you choose to compare the entire Yantra 7x configuration and deploy all resulting differences.
- New or empty databases - When using the tool to deploy data into a new or empty database, deploy the entire database.
- Foreign key interdependencies - Other scenarios with foreign key interdependencies between tables in different groups or other significant configuration changes may require you to perform a full deployment.
- Performance - Upon completion of the deployment operation, the CDT signals all servers in the Yantra 7x target environment to refresh locally-cached data. This is a resource-intensive and time-consuming operation and may cause slowdown in Yantra 7x until the cache reaches a steady state. To avoid these dips in the performance, deploy changes for the entire Yantra 7x configuration in one operation rather than multiple, small ones.

In addition, Yantra Corporation recommends scheduling routine deployment operations during periods of low activity.

14.3 Setting Up the Configuration Deployment Tool

Setting up the CDT involves installing software, copying files, and editing them as described in this section.

To set up the Configuration Deployment Tool:

1. If Yantra 7x has not been installed on your computer, install it by running the setup file in the CD-ROM appropriate for your operating system.

2. Copy the applicable JDBC driver file to the <YFS_HOME>/lib/ directory of your application server as follows:
   - If you are using Oracle and BEA WebLogic (JDK 1.4.*), copy the Oracle JDBC driver <ORACLE_HOME>/jdbc/lib/ojdbc14.jar file from your database server to the <YFS_HOME>/lib/ojdbc14.jar file on your application server machine.
   - If you are using Oracle and WAS (JDK 1.4.*), copy the Oracle JDBC driver <ORACLE_HOME>/jdbc/lib/ojdbc14.jar file from your database server to the <YFS_HOME>/lib/ojdbc14.jar file on your application server machine.
   - If you are using DB2, copy the <DB2_HOME>/sqllib/java/db2jcc.jar database driver file and the db2jcc_license_cu.jar license file on your database server to the <YFS_HOME>/lib/ directory.
   - If you are using SQL Server, copy the SQL Server JDBC driver file (Opta.jar) to your <YFS_HOME>/lib/ directory.
3. From the <YFS_HOME>/resources/ directory, copy and rename the following files by removing the .sample extension:
   - log4jconfig.xml.sample
   - yifclient.properties.sample
   - yfs.properties.sample
4. From the <YFS_HOME>/resources/ydkresources/ directory, copy and rename any files named .sample by removing the .sample extension. For example, rename ydkconfig.xml.sample to ydkconfig.xml.

   Note: Rename cdt_custom.xml.sample to cdt_custom.xml ONLY if you have custom tables to be included in the compare function.

5. If you have extended the Yantra 7x database, copy your database extensions to your computer as follows:
   a. Copy your database extension XML files to your local <YFS_HOME>/database/entities/extensions/ directory.
   b. Copy your yfsdbextn.jar file to your local <YFS_HOME>/lib/ directory.

For more information on extending your database see the *Yantra 7x Customization Guide*.

6. In the following files, specify the values described in Table 14–2:
   - <YFS_HOME>/resources/yfs.properties
   - <YFS_HOME>/resources/ydkresources/ydkconfig.xml
   - <YFS_HOME>/bin/ydk.cmd
   - <YFS_HOME>/resources/log4jconfig.xml

Table 14–2 Configuration Deployment Tool Properties

Property	Description
In the yfs.properties resource file	
yfs.dblogin.jdbcurl	– If you are using Oracle, set this value to: jdbc:oracle:thin:@<DatabaseServerHostname/IPaddress>:<TNSListenerPortNumber>:<DatabaseSID>. – If you are using SQL Server, set this value to: jdbc:inetdae7:<Database Server HostName/IPaddress>:<Port Number>?<database=Database Name>&charset=<your charset>. – If you are using DB2, set this value to: jdbc:db2://<DatabaseServerHostname>:<Port Number>/<Database name>.
yfs.dblogin.userid	Specify the user name associated with the database.
yfs.dblogin.password	Specify the password for the user associated with the database.
In the ydkconfig.xml resource file	
ResourcesDir	Specify as the full path of the <YFS_HOME>/resources/ydkresources/ directory, including the final slash. For example, ResourcesDir=D:/Yantra/resources/ydkresources/.
In the ydk.cmd script	
CLASSPATH	Include the <JAVA_HOME>/bin/ directory.
YFS_HOME	Specify the Yantra 7x installation directory. For example, YFS_HOME=D:/Yantra.

Table 14–2 Configuration Deployment Tool Properties

Property	Description
JAVA_HOME	Specify the Java installation directory. For example, `JAVA_HOME=D:/JDK_1.4.2_04_b05.`
DB_EXTN_JAR	Conditional. If you have extended any Yantra 7x configuration database tables, specify the full path to your `yfsdbextn.jar` file.
DB_DRIVER	Specify the path to the database driver. For example, `DB_DRIVER=D:/Yantra/extn/ojdbc14.jar`
In the log4jconfig.xml script	
LOG_DIRECTORY	Set this value to: `<param name="File" value="D:/Yantra/Yantra_Application.log">.`

14.4 Running the Configuration Deployment Tool

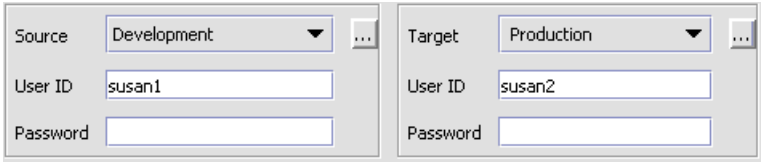
This section describes how to start and stop the Configuration Deployment Tool.

To start the Configuration Deployment Tool:

1. From the `<YFS_HOME>/bin/` directory, run the `ydk.cmd` script, which opens a Microsoft Windows console and starts the Yantra 7x Development and Deployment WorkBench.

   Important: The Windows console displays WorkBench startup information. ***Do not*** close the console while the WorkBench is running. Closing the console closes the tool, and your work is lost.

2. From the Yantra 7x Development and Deployment WorkBench menu, choose Tools > Deployment > Configuration Data Deployment. This opens the Configuration Deployment Tool Logon dialog box.



3. Choose the Source button and enter the values appropriate for the source database. Then choose the Target button and enter the values appropriate for the target database.

   When you are finished, close the dialog box. The values you specified are saved automatically and persist from one session to the next.

In the dialog boxes for the Source and Target databases, specify the applicable values as described in Table 14–3:

Table 14–3 Configuration Deployment Tool Logon Dialog Box

Field	Detail
Name	Specify a logical database identifier. For the source, specify the database you want to copy data from. For the target, specify the database to write the data to.
className	Specify the class name of your database driver as follows: • If you are using Oracle, set this value to: `oracle.jdbc.OracleDriver.` • If you are using SQL Server, set this value to: `com.inet.tds.TdsDriver.` • If you are using DB2, set this value to: `com.ibm.db2.jcc.DB2Driver`.

Table 14–3 Configuration Deployment Tool Logon Dialog Box

Field	Detail
jdbcURL	Specify the URL to connect to the database: • If you are using Oracle, set this value to: `jdbc:oracle:thin:@<DatabaseServerHostname/IP address><TNSListenerPortNumber>:<DatabaseSID>.` • If you are using SQL Server, set this value to: `jdbc:inetdae7:<Database Server HostName/IPaddress>:<Port Number>?<database=Database Name>&charset=<your charset>.` • If you are using DB2, set this value to: `jdbc:db2://<Database Server Hostname>:<Port Number>/<Database name>.`
dbType	Specify the type of database you are running as follows: • For Oracle, specify `oracle` (all lower case). • For SQL Server, specify `sqlserver` (all lower case). • For DB2, specify `db2` (all lower case). • For an XML datasource, specify `xml` (all lower case).
folder	If using an XML datasource, specify the complete path of the folder location for the XML files.
httpurl	Only applicable for the target database. Specify a URL for the application server whose data cache is to be refreshed after data is deployed into the target database. Use the syntax: `http://<hostname/ip-address>:<port-number>/yantra/interop/InteropHttpServlet`, where hostname, IP-address and port-number are the parameters used to connect to the application server.
schema	Specify the schema owner as follows: • If you are using Oracle or DB2 database and the user you specify is different from the Yantra 7x schema owner, specify the owner of the Yantra 7x schema. • If you are using SQL Server, leave this blank.
user	Specify the user name associated with the database.

4. In the Logon dialog box, enter the passwords associated with the user names.

   The Deployment Explorer window displays.

To stop the Configuration Deployment Tool:

From the Yantra 7x Development and Deployment WorkBench menu, choose File > Exit.

This closes the Configuration Deployment Tool and the Windows console.

14.5 Understanding the Configuration Deployment Tool User Interface

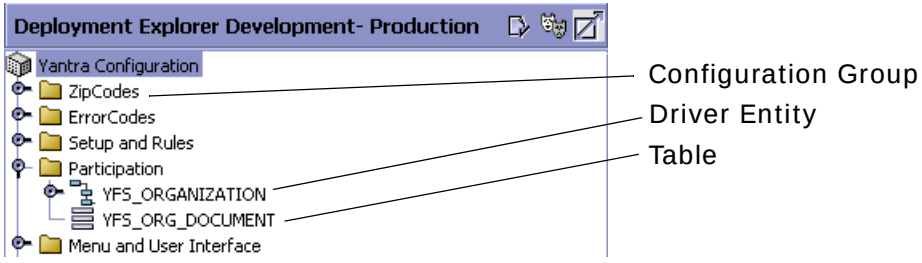
When the Configuration Deployment Tool starts, it prompts you to specify the details about your source and target databases to use during the session. After you have successfully connected to your source and target databases, the Deployment Explorer window appears.

14.5.1 The Deployment Explorer

The Deployment Explorer window displays the list of configuration groups, driver entities, and tables that can be deployed. The names you define for the source and target databases are displayed in the heading panel.

Each time you log into the Configuration Deployment Tool there is one instance of this window.

Figure 14–1 Deployment Explorer Window



You can choose the configuration group or the driver entity that you want to compare between the source and target databases.

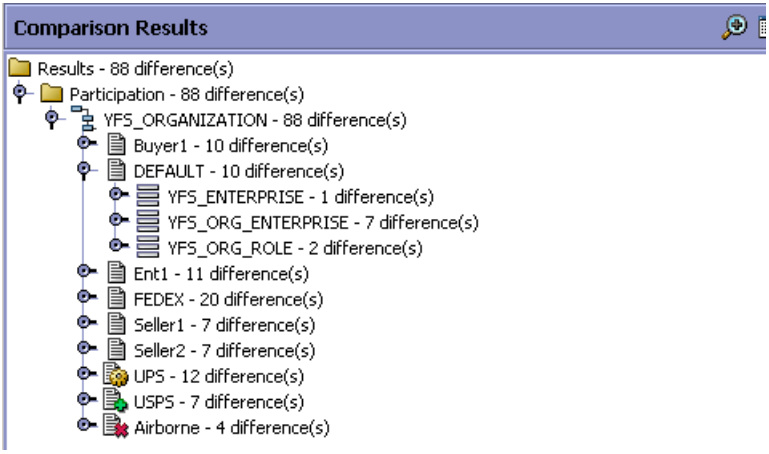
During the compare operation, the progress and the results of the comparison operation are displayed in the Comparison Results window and in the Status panel.

14.5.2 Comparison Results Window

The Comparison Results window displays the outcome of the comparison between the source and the target databases.

The Comparison Results window displays information pertaining to the current session. Only one Comparison Results window can be displayed during each session. After viewing the results of one comparison, you must close the window before you can compare a different set of tables.

Figure 14–2 Comparison Results Window



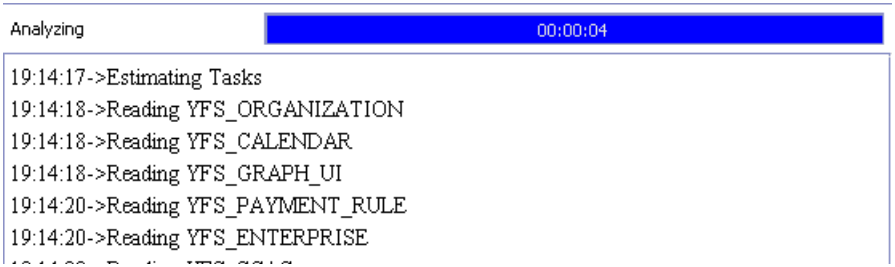
After generating comparison results, you can carry out any one of the following tasks:

- Generate a report of the differences
- View the details of each difference
- Deploy configuration data from the source database into the target database

14.5.3 The Status Panel

The Status panel displays information about operations while they are carried out.

Figure 14–3 Configuration Deployment Tool Status Panel



14.6 Specifying the Preferences Settings

You can configure preferences (such as a reports directory) and parameters that determine the behavior of the comparison operation. When you modify these properties, the changes persist, so you do not need to reset them each time you use the CDT. These changes are saved in the `<YFS_HOME>/resources/ydkresources/ydkprefs.xml` file.

To specify Configuration Deployment Tool settings:

1. From the Deployment Explorer action bar, choose the *Preferences* icon.
2. In the Preferences window, fill in values using the descriptions in Table 14–4, "Configuration Deployment Tool Preference Settings".

Table 14–4 Configuration Deployment Tool Preference Settings

Control	Description
Settings Tab	
Reports Directory	Specify the absolute path where you want reports to be generated.
Custom Deployment Class	Specify the name of the class that should be invoked for deploying custom tables not handled by the CDT.

Table 14–4 Configuration Deployment Tool Preference Settings

Control	Description
Max Changes to Display	Specify the maximum number of differences to be displayed. The default display number is 100.
Transformations Tab	
Table Element	Tables that can be added or deleted.
Table Name Attribute	Specify the name of the table on which you want to carry out the transformation. The syntax and case must match the name of the table used in the Yantra 7x ERDs. Custom tables cannot be transformed. Choose the Details icon to specify a value.
Column Element	Columns that can be added or deleted.
Column Name Attribute	Specify the name of the column containing the data to be transformed. The syntax and case must match the name of the column used in the Yantra 7x ERDs. Extended columns can be transformed. Choose the Details icon to specify a value.
Transform Element	Define the transformation for this column. For each column, you can define one or more transformations. These transformations are applied to data in this column in sequential order. You can specify multiple transformations for each column, using the delete action to remove the parent element.
Match Attribute	Specify the pattern to search for in the source data. All matching occurrences of this pattern are replaced with the value specified in the Replace attribute. Choose the Details icon to specify a value.
Replace Attribute	Specify the value to replace the pattern with. Choose the Details icon to specify a value.
XPath Attribute	Conditional. If the column to be transformed contains non-XML data, you do not need to specify this XPath attribute. However, some configuration information in Yantra 7x is stored as XML in the database. If the column to be transformed contains XML data, use this attribute to specify the location of the exact attribute to be transformed. Use the syntax: `xml:/Configuration/Connection/Host/@IPAddress`. Choose the Details icon to specify a value.
Append-only Tables Tab	

Table 14–4 Configuration Deployment Tool Preference Settings

Control	Description
Append-only Tables	Specify any configuration tables in which *some* rows maintain data that is external to Yantra 7x. This prevents the data from being deleted during deployment. Specify that table **and** all of its dependent tables. **Note**: Rows that are maintained externally should ***never*** be present in your source database, since this can lead to unpredictable results.
Ignore Tables Tab	
Ignore Tables	Specify any external configuration tables that you do **not** want the tool to deploy from the source to the target. Ignoring a table automatically ignores all dependent tables as well.

14.7 Transforming Elements

When deploying data from one database instance to another, you can override the values of certain data elements. For example, if your source and target environment network settings (host names, port numbers, and IP addresses) are different, the Configuration Deployment Tool can transform the settings in order to make them appropriate for the target environment.

Transformations are carried out as a pattern match and replace the data in the source database before it is deployed into the target.

> **Note:** The match and replace are carried out for the complete string literal and no wild card search for characters is allowed.

Example Transformation

For example, consider the following configuration XML in the source database:

```
<SubFlowConfig>
  <Link>
    <Properties DeliveryMode=""
     InitialContextFactory="weblogic.jndi.WLInitialContextFactory"
     ProviderURL="t3://localhost:7001" QCFLookup="TEST_AGENT_QCF"
```

```
     QName="DefaultAgentQueue" TimeToLive=""/>
   </Link>
</SubFlowConfig>
```

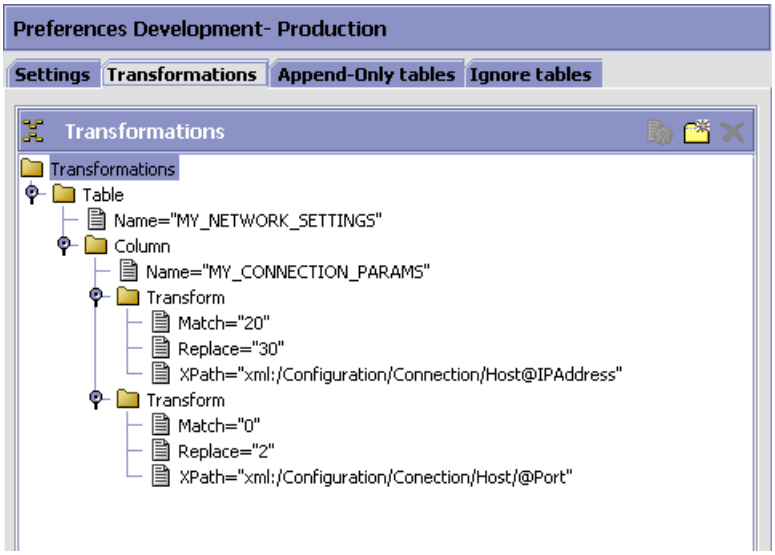
The target database has values for port number as 7221 and QCFLookup as AGENT_QCF, which you do not want overridden by values in the source database. To transform these values, specify the values as described in Table 14–5:

Table 14–5 Transforming Elements

Element	Attribute	Value
ProviderURL		
	Match	Specify 7001 (or 0 to find all occurrences of 0).
	Replace	Specify 7221 (or 2 to replace all occurrences of 2).
	XPath	`xml:/SubFlowConfig/Link/Properties/@ProviderURL`
QCFLookup		
	Match	Specify TEST_ to find all occurrences of TEST_.
	Replace	Leave blank to ensure that TEST_ is removed.
	XPath	`xml:/SubFlowConfig/Link/Properties/@QCFLookup`

Using this example, the Transformation tab would look as shown in Figure 14–4, "Transformations Example".

Figure 14–4 Transformations Example



To transform elements of the configuration data:

1. From the Deployment Explorer window action bar, choose the *Preferences* icon.
2. In the Preferences window, select the Transformations tab and fill in values, using the information provided in Table 14–4, "Configuration Deployment Tool Preference Settings" on page 158. When you deploy data, these transformation values you specify are deployed along with configuration data.

   Before you deploy data, you must first perform a database comparison as described in Section 14.8, "Comparing Data".

14.8 Comparing Data

In order to deploy configuration data into production, you must first compare the two databases and then deploy your changes.

To compare databases:

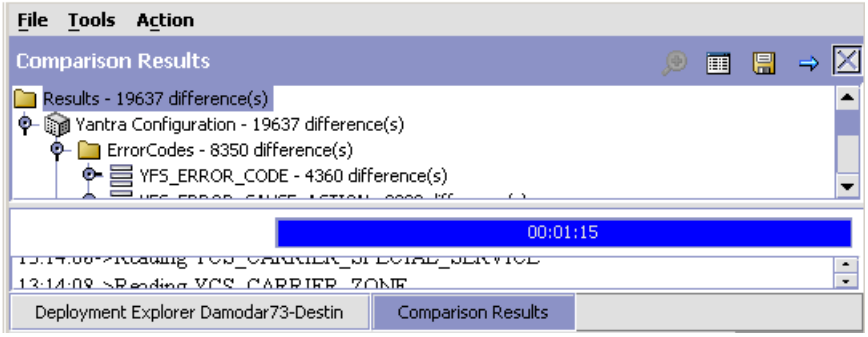
1. From the Deployment Explorer tree, select the data you want to compare.

Tip: When you are deploying data for the ***first*** time, compare the entire database. Compare smaller increments only ***after*** you are certain that your source and target databases have relatively few differences.

The available comparison groups are as follows:

- For the entire database, choose the *Yantra 7x Configuration* icon.
- For a specific configuration group, choose the *Configuration Group* icon.
- For a specific driver entity, choose the *Driver Entity* icon.

2. From the Deployment Explorer action bar, choose the *Compare* icon. The Comparison Results window displays on the top right and lists all differences. The Comparison Results Status panel displays on the bottom right.



Note: If a table is present in multiple groups or under multiple entities, its difference may be counted multiple times in the total number.

After the comparison completes, you can perform any of the following actions:

- Examine any differences, using the instructions in Section 14.9, "Examining Differences Between Databases" on page 164.

- Export the report, using the instructions in Section 14.10, "Exporting Comparison Results" on page 165.
- Generate a report, using the instructions in Section 14.12, "Generating a Report of Differences" on page 167.
- Deploy your changes, using the instructions in Section 14.13, "Deploying Your Configuration Data" on page 168.

14.9 Examining Differences Between Databases

You can examine the details of each database record difference.

To examine differences between databases:

1. Compare the two databases (by choosing the *Compare* icon). For detailed procedures, see Section 14.8, "Comparing Data" on page 162.
2. From the Comparison Results tree, expand an entity and select the table that you want to examine.
3. From the Comparison Results action bar, choose the *Details* icon. The range of possible results are as follows:

   Unchanged icon - indicates that an entity contains dependent tables that have differences (these tables may indicate inserts, deletes, updates, or a combination).

   Add icon - indicates that a record will be inserted to the target database, as shown below.

Record Details

Inserted

Name	Value
OrganizationCode	USPS
CatalogOrganizationCode	DEFAULT
PrimaryUrl	

Remove icon - indicates that a record will be deleted from the target database, as shown below.

Record Details

Deleted

Name	Value
OrganizationCode	Airborne
CatalogOrganizationCode	DEFAULT
PrimaryUrl	

Modify icon - indicates columns that will be updated on the target database as shown below. It displays information in two sections:

- Top section - values that will be changed on the target database
- Bottom section - values that will remain the same

Record Details

Changed

Name	Old Value	New Value
PaymentProcessingReqd	Y	N

Unchanged

Name	Value
OrganizationCode	UPS
CatalogOrganizationCode	DEFAULT
PrimaryUrl	

4. After examining your data, you may want to generate a report of these differences as described in Section 14.12, "Generating a Report of Differences" on page 167.

14.10 Exporting Comparison Results

You can export the configuration differences for comaprison at a later time or as a backup for you existing configuration.

To export the comparison results into an XML file:

1. Make sure you have specified the `reports` directory. For more information about specifying preferences settings, see Section 14.6, "Specifying the Preferences Settings" on page 158.
2. From the Comparison Results action bar, choose .
3. From the Windows Explorer, browse to your `reports` directory. In this directory, the Configuration Deployment Tool automatically creates a subdirectory. For example, if you have specified `D:/reports` and exported the comparison results at 3:40 pm on May 23, 2003, the CDT creates a subdirectory as: `D:/reports/export20030523154024`. This new subdirectory contains the file `ydkexport.xml` that represents the comparison results.

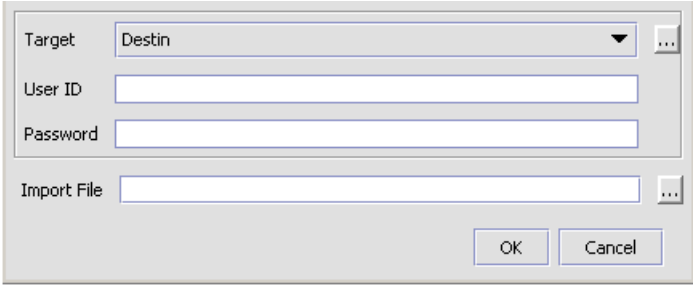
14.11 Importing Configuration Differences

You can import configuration differences that are obtained by exporting comparison results.

For more information about exporting comparison results, see Section 14.10, "Exporting Comparison Results" on page 165.

To import configuration differences:

1. Run the `<YFS_HOME>/bin/ydk.cmd` script. This script opens a Microsoft Windows console and starts the Yantra 7x Development and Deployment WorkBench.
2. From the Yantra 7x Development and Deployment WorkBench menu, choose Tools > Deployment > Import Results. The Configuration Deployment Tool Import dialog box displays.



3. Choose the Target button and enter the values appropriate for the Target database. Then choose the Import File and enter the path of the file to be imported.

   When you are finished, close the dialog box by clicking the OK button.

4. After the comparison results are loaded, you can perform any of the following actions:

   - Examine any differences, using instructions as specified in Section 14.9, "Examining Differences Between Databases" on page 164.
   - Generate a report, using the instructions as specified in Section 14.12, "Generating a Report of Differences" on page 167.
   - Deploy your changes, using instructions as specified in Section 14.11, "Importing Configuration Differences" on page 166.

14.12 Generating a Report of Differences

You can generate a report of differences between the source and target databases.

To generate a report of differences:

1. Ensure that you have specified a reports directory (by choosing the *Preferences* icon). For detailed information about preference settings, see Section 14.6, "Specifying the Preferences Settings" on page 158.
2. Compare the two databases (by choosing the *Compare* icon). For this detailed procedure, see Section 14.8, "Comparing Data" on page 162.
3. In the Comparison Results tree, select the Results node.
4. From the Comparison Results action bar, choose the *Generate Report* icon. The Status panel displays trace messages that enable you to determine the success of the report generation process. The location of the report displays along with a message of the successful creation of reports.

5. From the Windows Explorer, browse to your reports directory. Within the directory you specified, the Configuration Deployment Tool creates a subdirectory named according to the time it was created.

   For example, if you have specified `d:/reports` as the reports directory and generate a report at 3:40 p.m. on May 23, 2003, the CDT creates a subdirectory called `20030523154024` in the `d:/reports` directory.

   This new subdirectory contains the following:

   - An `index.xml` file contains an overall summary of changes as displayed on the UI
   - One XML file for each table that has changes with the details of each change

6. Open the XML files to see the differences.

If you generate another report, a new directory is created and populated with another set of XML files.

14.13 Deploying Your Configuration Data

Before deploying configuration data, ensure that you are deploying the correct data by comparing the data and examining any differences.

In addition, ensure that you have addressed rollback issues. Yantra 7x supplies sample backup and rollback scripts. For information on these scripts, see Section 14.15.4, "Data Rollback Scripts" on page 174.

To deploy configuration data:

1. Compare the two databases (by choosing the *Compare* icon). For this detailed procedure, see Section 14.8, "Comparing Data" on page 162.
2. From the Comparison Results tree, select the entities you want to deploy.

Tip: When you deploy data for the ***first*** time, deploy the entire database. Deploy smaller increments only ***after*** you are certain that your source and target databases have relatively few differences.

The available options are as follows:

- For the entire database, choose the *Yantra 7x Configuration* icon.
- For a specific configuration group, choose the *Configuration Group* icon.
- For a specific driver entity, choose the *Driver Entity* icon.

3. From the Comparison Results action bar, choose the *Deploy* icon.
   - If the deployment succeeds, the Status panel displays a success message, the data is committed to your target database, and the cache is updated as specified in the `httpurl` field described in Table 14–3.
   - If the deployment fails, the Status panel indicates the errors to resolve and no data is committed to the target database.
4. If you have deployed data from the YFS_RESOURCE table, restart your application servers in the target environment in order to refresh the cache.

14.14 Deploying Your Configuration Data in Command-Line Mode

There may be circumstances under which you want to run or schedule deployment of configuration data without user interface interaction or without viewing the source and target comparison results.

To accomplish this you can deploy your configuration data in command-line mode. When you deploy your data in command-line mode, the CDT automatically compares the source and target environments and then deploys the configuration data.

To deploy configuration data in command-line mode:

1. Set up the CDT following the instructions in Section 14.3, "Setting Up the Configuration Deployment Tool" on page 150.
2. Run the CDT.
3. Set the properties in the <YFS_HOME>/bin/cdtshell.cmd file as described in Table 14–6.

Table 14–6 Configuration Deployment Tool Properties

Property	Description
JAVA_HOME	Specify the Java installation directory. For example, JAVA_HOME=D:/JDK_1.4.2_04_b05.
YFS_HOME	Specify the Yantra 7x installation directory. For example, YFS_HOME=D:/Yantra.
DB_DRIVER	Specify the path to the database driver. For example, DB_DRIVER=D:/Yantra/extn/ojdbc14.jar
DB_EXTN_JAR	Conditional. If you have extended any Yantra 7x configuration database tables, specify the full path to the yfsdbextn.jar file.
SOURCE_DB	Specify the name of the data source as defined in the Configuration Deployment Tool Logon dialog box.
SOURCE_PASSWORD	Conditional. If using a database as the source destination for data, specify the password for the database instance.
TARGET_DB	Specify the name of the data target destination for the data. For example, TARGET_DB=xxx (if using a database) or TARGET_DB=xxx (if using an XML file).
TARGET_PASSWORD	Conditional. If using a database for the target destination, specify the password for the database instance.
ExportDir	The comparison results are stored in this directory.
ImportDir	This directory should contain the comparison results that are exported. The ImportDir property when present, the source database properties are not used.
DoNotSynchronize	Valid values are Y or N. Y indicates that the comparison results are exported, but are not deployed. By default, the comparison results are automatically deployed.

4. Run the <YFS_HOME>/bin/cdtshell.cmd script.

You can also schedule this script to run at any appropriate time.

14.15 Troubleshooting

During operations, the Configuration Deployment Tool displays messages in the Status panel that enable you to understand the status of each operation. These messages can be classified as:

- Status
- Warnings
- Unexpected errors

The following section describes various messages that you may encounter and any relevant corrective actions you should take.

14.15.1 Informational Messages

Informational messages represent the status of the operation being performed. These messages are displayed in the default color (typically black) in the Status panel. Examples of informational messages include:

- Refreshing database cache
- Deployment operation started
- Reading table YFS_ORGANIZATION

14.15.2 Warning Messages

Warning messages typically require corrective action. They are displayed in red on the Status panel. The CDT may produce the warning messages described in this section.

WARNING - FK check failed for table <name> to <name2>

This warning message typically indicates that the configuration data that you are trying to deploy will cause inconsistent data in the target database.

To analyze and correct this problem:

1. Determine the size of the data set you are deploying. This error typically occurs when trying to deploy a very small set of data, such as only a driver entity or a configuration group. For example, when deploying a pipeline, this error results if the document type to which the pipeline belongs has not been picked for deployment.

   Try resolving this error by selecting a larger set to deploy. For example, instead of deploying a record, deploy the entire group, if possible.

2. If you still encounter this error for a group or you must only deploy a particular record, try synchronizing the foreign table before deploying the data.

3. Occasionally, inconsistent data in the source database causes this error. If this is the case, you must correct the source of the inconsistency before you proceed.

WARNING - Cache Refresh Failed

This error indicates that the CDT was unable to inform the application server cluster on the target environment about the newly deployed configuration changes. The reason the cache refresh failed is displayed on the Status panel.

To analyze and correct this problem:

1. Verify the URL specified in the `httpurl` field for the target database. The `httpurl` is accessible from the Logon dialog box. Ensure that the `httpurl` points to a running instance of the application server and has the following format:

   `http://<hostname/ip-address>:<port-number>/yantra/interop/InteropHttpServlet` where hostname, ip-address and port-number are the parameters used to connect to the application server.

2. If your target environment is not running, no action is required. Yantra 7x automatically reads the latest configuration data when it is started.

3. If the target environment is running, you must manually drop the stale database cache using the Yantra 7x System Administration Console. Not performing this step may result in Yantra 7x not recognizing the changed configuration.

WARNING - The program detected a few abandoned records in the target database.

In most cases, abandoned records are harmless and should not lead to the incorrect operation of Yantra 7x. By default, the CDT leaves them untouched.

This warning typically occurs as a result of the following circumstances which are described in detail in Section 14.1.3, "Externally Maintained Configuration Data" on page 142:

- When the CDT finds records that do not belong to a valid driver entity (for example, a pipeline for a process type that no longer exists)
- When the CDT has been configured to ignore certain tables without ignoring all dependent tables

To analyze and correct this problem:

1. Add the `-DShowAbandoned=Y` Java parameter to the `ydk.cmd` script.
2. Run the `ydk.cmd` script. If the CDT finds abandoned records, it dynamically creates a group called "Abandoned Records" and displays them in the Comparison Results window.
3. Examine these records, and then either ignore them or delete them from the target.

14.15.3 Unexpected Errors

Depending on the severity, messages about unexpected errors are displayed in either of the following places:

- In the CDT Status panel (in red)
- In the Microsoft Windows console used to launch the Configuration Deployment Tool

To analyze and correct these errors:

1. If the error indicates an out-of-memory condition, try your previous operation with a smaller set of data.
2. Verify that your system specifications comply with the recommendations described in Chapter 2, "System Requirements" on page 3.

3. You can edit the `-mx` Java parameter in the `ydk.cmd` script to increase the memory available for the Configuration Deployment Tool.

For example, if you were comparing the complete configuration, try comparing one group at a time. The same is true for the deployment operation.

In other cases, the underlying error and detailed trace are displayed. This may point to an incomplete or faulty installation or incorrectly specified runtime parameters.

14.15.4 Data Rollback Scripts

Before deploying data from a staging to a production environment, it is recommended to take a snapshot of your production configuration data. This snapshot enables you to perform a rollback of the deployment operation in case of failure. Yantra 7x provides the following rollback scripts:

- Backup script - creates multiple files containing data from all Yantra 7x configuration data
- Restoration script - uses the files produced by the backup scripts to restore the Yantra 7x configuration to a previously known good state

> **Note:** The backup files do not represent the entire configuration snapshot (for example, it does not capture the YFS_PERSON_INFO table), so do not use the scripts for deploying data to a different database. Instead, use the CDT to deploy configuration data.
>
> These scripts are ***only*** for performing a rollback of configuration data onto the database from which the snapshot was taken.

14.15.4.1 Customizing the Scripts

Yantra 7x supplies sample scripts in the `<YFS_HOME>/bin/` directory. Rename and customize them to suit your implementation and methodology. Some suggested changes include adding your custom configuration tables and changing the paths where the data files are written to or read from to suit your version control process.

These scripts rely on utilities provided by the database vendor.

Oracle scripts rely on `export` or `import` or `sqlplus`. Edit and use the following scripts:

- `backup_config_oracle.cmd`
- `restore_config_oracle.cmd`

SQL Server scripts rely on `bcp` or `osql`. Edit and use the following scripts:

- `backup_config_sqlserver.cmd`
- `restore_config_sqlserver.cmd`

DB2 scripts rely on `export` or `load`. Edit and use the following scripts:

- `backup_config_db2.cmd`
- `restore_config_db2.cmd`

14.15.4.2 Running the Scripts

Before you deploy any data using the Configuration Deployment Tool, use the backup script to back up your data. These backup data files can then be version controlled.

A

Product Licensing and Packaging

This appendix provides the Yantra 7x licensing agreement, a brief description of each of the product components, and a description of each of the Yantra 7x applications.

A.1 Yantra 7x Software License Agreement

Use of any Yantra 7x component or application assumes acceptance of the following license agreement during the software installation procedure:

This software (the "Software") is the copyrighted work of Yantra Corporation ("Yantra") and/or its suppliers. The Software is made available solely for use by licensed users according to the license agreement between Yantra, or Yantra's authorized distributors or resellers, and such licensed user (the "License Agreement"). Use of the Software, including any application therein or portion thereof, is governed by the terms and conditions of the License Agreement. Any use, reproduction or redistribution of the Software not in accordance with the License Agreement or by an unlicensed party is expressly prohibited by law.

By clicking "Accept", you acknowledge and agree that you and/or your employer (i) are a licensed user of the Software, (ii) have executed a License Agreement for the use of the Software, and (iii) will only use the Software, including any application therein or portion thereof, as permitted by the License Agreement. If you do not agree with the foregoing sentence, click "Do Not Accept" and the installation process will stop.

WITHOUT LIMITING THE FOREGOING, ANY REPRODUCTION OF THE SOFTWARE TO ANY OTHER LOCATION OR SERVER FOR USE, FURTHER REPRODUCTION OR DISTRIBUTION IS EXPRESSLY PROHIBITED, UNLESS SUCH USE, REPRODUCTION OR DISTRIBUTION IS EXPRESSLY PERMITTED BY THE LICENSE AGREEMENT.

THE SOFTWARE IS WARRANTED ONLY ACCORDING TO THE TERMS OF THE LICENSE AGREEMENT.

EXCEPT AS EXPRESSLY STATED IN THE LICENSE AGREEMENT, YANTRA MAKES NO EXPRESS OR IMPLIED REPRESENTATIONS OR WARRANTIES WHATSOEVER REGARDING THE SOFTWARE OR OTHERWISE. YANTRA DISCLAIMS ALL WARRANTIES, WHETHER EXPRESS, IMPLIED OR STATUTORY, INCLUDING, BUT NOT LIMITED TO, WARRANTIES OF THE MERCHANTABILITY, TITLE, NON-INFRINGEMENT, FITNESS FOR ANY PARTICULAR PURPOSE, OR ARISING FROM A COURSE OF DEALING, USAGE OR TRADE

RESTRICTED RIGHTS LEGEND.

If the licensed user of the Software is the United States of America, its agencies and/or instrumentalities ("U.S. Government"), or the license granted under the License Agreement is pursuant to a contract with the U.S. Government, the Software is provided with Restricted Rights. Use, reproduction, or disclosure by or on behalf of the U.S. Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 or subparagraphs (c)(1) and (2) of the Commercial Computer Software - Restricted Rights at 48 CFR 52.227-19, as applicable, or in successor provisions, as well as the limitations set forth in the License Agreement.

Manufacturer is Yantra Corporation, One Park West, Tewksbury, MA 01876.

Any rights not expressly granted in the License Agreement are reserved.

A.2 Yantra 7x Components

Yantra 7x is comprised of the following components:

- Yantra 7x Application Consoles - the user interface that provides the application functionality.
- Yantra 7x Configurator - the user interface that provides Yantra 7x product and business process configuration functionality.
- Yantra 7x Platform - the enabling foundation for all Yantra 7x applications. It provides the differentiating capabilities that are required for managing processes in an extended enterprise environment. These capabilities include the:
  - Multi-Enterprise Data Model
  - Participant and Relationship Model

- Yantra 7x Service Definition Framework (which enables the flexible configuration of key processes utilizing business rules, events, workflow and collaboration)
- Interoperability Services
- Yantra 7x Presentation Framework (for deploying application functionality through Yantra 7x user interfaces, portals, wireless and XML/EDI).

Additionally, the Yantra 7x Platform includes the application framework and runtime environment, as well as an extensibility and deployment workbench.

A.3 Yantra 7x Platform

With your purchase of Yantra 7x, you received the Yantra 7x Platform and are licensed to use the following Yantra 7x functional components:

- Product Management
- Participant Model
- COGNOS Data Dictionary
- Alert/Exception Console
- Service Definition Framework
- Label Print Infrastructure
- Other infrastructure components in Platform utilized by Yantra 7x applications such as Security Framework, UI Framework, Service Definition Framework

A.4 Yantra 7x Applications

The Yantra 7x Applications include:

- Yantra 7x Distributed Order Management
- Yantra 7x Supply Collaboration
- Yantra 7x Logistics Management
- Yantra 7x Reverse Logistics
- Yantra 7x Inventory Synchronization

- Yantra 7x Delivery and Service Scheduling
- Yantra 7x Warehouse Management System

A.4.1 Yantra 7x Distributed Order Management

Yantra 7x Distributed Order Management manages and coordinates the entire customer order lifecycle (from order capture to shipment) at the line-item level. It provides Process Modeling capabilities for order creation, visibility, order tracking and modification to the seller of the order. It utilizes Process Modeling capabilities, to process and source orders according to their unique characteristics.

If you purchased the Distributed Order Management application, you are licensed to use the following Yantra 7x components:

- Order Console
- Create Order
- Outbound Shipment Console
- Create Outbound Shipment
- Work Order Console (for MTO Work Orders)
- Order Sourcing and Scheduling

Processes for the following documents can be managed through this application.

- Sales Order
- Transfer Order
- Template Order
- Drop Ship Purchase Order

A.4.2 Yantra 7x Supply Collaboration

Yantra 7x Supply Collaboration manages and coordinates the entire lifecycle of supply transactions at the line-item level. It provides capabilities for planned order creation, visibility and evolution into actual purchases. Purchases can be tracked, monitored, and received to generate a complete and real-time view of all on-going supply transactions.

If you purchased the Supply Collaboration application, you are licensed to use the following Yantra 7x components:

- Inbound Order Console
- Inbound Shipment Console
- Create Inbound Order
- Receipt Console

Processes for the following documents can be managed through this application.

- Purchase Order
- Planned Order

A.4.3 Yantra 7x Logistics Management

Yantra 7x Logistics Management extends our execution capabilities beyond order management. It provides an implementation with the tools to define and manage complex logistics processes, including such best-in-class practices as merge-in-transit and consolidation and de-consolidation of multi-leg and multi-mode shipments. Process definition and execution by shipment and by load, monitoring based on shipment-specific dates and visibility into on-going status for all participants in the delivery chain are key benefits of this application.

If you purchased the Logistics Management application, you are licensed to use the following Yantra 7x components:

- Plan Console
- Inbound Shipment Console
- Outbound Shipment Console
- Load Console
- Routing

Processes for the following documents can be managed through this application.

- Purchase Order Shipments
- Transfer Order Shipments
- Sales Order Shipments

A.4.4 Yantra 7x Reverse Logistics

Yantra 7x Reverse Logistics provides the tools to model and manage complex reverse logistics operations. The Reverse Logistics application can be implemented to manage the reverse logistics process for orders shipped from within the Yantra 7x Distributed Order Management application or for orders shipped with other order management applications. Complete reverse logistics management processes include return authorization, multi-step receiving and disposition, credit processing, and processing of exchange and refurbishment orders.

If you purchased the Reverse Logistics application, you are licensed to use the following Yantra 7x components:

- Return Console
- Create Return
- Order Console
- Recepit Console

Processes for the following documents can be managed through this application.

- Return Order
- Exchange Order
- Return to Vendor

A.4.5 Yantra 7x Inventory Synchronization

Yantra 7x Inventory Synchronization manages and coordinates global inventory across multiple sites, enterprises and participants. It allows the sharing of inventory levels with external systems for reconciliation and collaboration purposes.

If you purchased the Inventory Synchronization application, you are licensed to use the following Yantra 7x components:

- Inventory Console
- Adjust Inventory
- Available to Promise
- Work Order Console for Build-to-Stock Work Orders

- Inventory Costing

A.4.6 Yantra 7x Delivery and Service Scheduling

Yantra 7x Delivery and Service Scheduling provides you with the means to model and manage extended product and service fulfillment networks. Built around the Yantra 7x core extended enterprise data model and business process modeling, these components provide the functionality and tools to capture all product and service requests and present a coordinated picture of fulfillment and execution to any end user.

If you purchased the Delivery and Service Scheduling application, you are licensed to use the following Yantra 7x user interface components:

- Capacity Console
- Work Order Console
- Service Sourcing and Scheduling

A.4.7 Yantra 7x Warehouse Management System

Yantra 7x Warehouse Management System is a highly scalable warehouse management system designed for high-volume pick, pack, personalize, and ship operations. It serves as a planning, execution and measurement tool for distribution managers, bringing operational discipline to complex warehouse operations. Because it leverages the Yantra 7x Platform, it can notify other systems or locations of critical events that occur within the warehouse and make fulfillment decisions based on real-time information received from customers, partners, and suppliers. Built to handle the most complex fulfillment requirements, it improves internal efficiencies through business rules-driven warehouse management.

The Warehouse Management System application is comprised of the following components:

- WMS Inbound
- WMS Inventory Management
- WMS Value Added Services (VAS)
- WMS Outbound
- Yantra 7x Carrier Server (YCS)

A.4.7.1 WMS Inbound

If you purchased the WMS Inbound component, you are licensed to use the following Yantra 7x user interface components:

(Everything under the WMS-Inbound Menu)

- Inbound Shipment Console
- Visibility to Inbound POs
- Create Inbound Shipment
- Create Inbound Load
- Load Console
- Receive
- Inspect Receipt
- UnReceive
- Mobile Receipt
- Return Shipment Console
- Return Receipt Console
- Create Return
- Inspect Return

A.4.7.2 WMS Inventory Management

If you purchased the WMS Inventory Management component, you are licensed to use the following Yantra 7x user interface components:

(Everything under WMS-Inventory Management Menu)

- Location Inventory Console
- Adjust Location Inventory
- Create Count Request
- Count Request Console
- Create Move Request
- Move Request Console
- Change Location Inventory Attributes

- Mobile Putaway/Retrieval
- Task Console
- Productivity Console
- Log Productivity
- Confirm Batch
- Global Inventory Visibility

A.4.7.3 WMS Value Added Services (VAS)

If you purchased the WMS Value Added Services (VAS) component, you are licensed to use the following Yantra 7x user interface components:

- Create Work Order
- Work Order Console
- VAS Station

A.4.7.4 WMS Outbound

If you purchased the WMS Outbound component, you are licensed to use the following Yantra 7x user interface components:

- Outbound Shipment Console
- Outbound Load Console
- Create Outbound Shipment
- Wave Console
- Create Wave
- Pack Station
- Manifest Station
- Manifest Console
- Load Console
- Mobile Pick/Pack Capabilities

A.4.7.5 Yantra 7x Carrier Server (YCS)

If you purchased the Yantra 7x Carrier Server component, you are licensed to use the following Yantra 7x functional components:

- FedEx Integration
- DHL Integration
- UPS Integration

A.4.8 Third-Party Licensing

Yantra 7x integrates with various third-party components. You must ensure appropriate licensing to use these third-party components. Yantra 7x integrates with the following third-party components:

- Yantra 7x Analytics
  - COGNOS PowerPlay
  - COGNOS FlexPack Bundle
  - CONGNOS ReportNet
- Loftware

  This product also includes software developed by the Apache Software Foundation (at http://www.apache.org/).

  For specific licensing information for an Apache product, see the Apache license agreement pertinent to the software version you have installed at: http://www.apache.org/licenses/LICENSE-1.1 for version 1.1 or

  http://www.apache.org/licenses/LICENSE-2.0 for version 2.0.
- JasperReports

  This product also includes the software developed by JasperSoft (http://www.jaspersoft.com). You should refer to their website for specifc licensing agreement.

Index

A

access control, 120, 129
Activesync, 69
agent criteria
  override, 102
agent server setting
  modifying configurator UI, 91
  modifying management.properties, 90
  modifying yfs.properties, 90
Apache server
  WebLogic proxy, WebSphere proxy, 114
  websphere proxy, 114
assigning ship nodes, 96
authentication
  using JAAS, 13
  using LDAP, 13

B

barcode printing
  WebLogic, 120
  WebSphere, 128
browser
  clearing cache, 137
bsfengines.jar, 117
bsf.jar, 117
buildWLS.xml, 121, 124
buildWS.xml, 129, 131

C

capacity planning, 50
CDT (Configuration Deployment Tool), 140
clearing browser, 136
clearing java plugin caches, 136
client character, 136
  displaying special characters, 136
client characters
  displaying unicode character, 136
cold boot
  reinstallation, 71
collation property, 49
component object model., 18
COM+ (extended component object model), 18
Configuration Deployment Tool, 139
  and externally maintained data
    appending, 143
    best practices, 142
    data integrity, 150
    ignoring, 143
  custom tables, 150
    best practices when using, 150
    specifying, 158
  data transformations, 145, 160
    specifying, 159
  exporting
    comparison results, 166
  features, 140
  foreign key interdependencies, 144, 150
    best practices, 150
    troubleshooting, 172
  groups
    configuration groups, 142
    driver entities, 141
  importing
    configuration differences, 166

performance
optimizing, 150
time estimates, 145
planning
change management strategy, 147
prerequisites, 146
rollback strategy, 147
security, 146
system requirements, 145
upgrades, 147
tasks
comparing databases, 162
deploying data, 168
deploying data in command-line mode, 170
examining differences between databases, 164
generating a report of differences, 167
performing rollback, 174
setting up the CDT, 150
specifying a reports directory, 158
specifying custom classes, 158
specifying tables to append, 159
specifying tables to ignore, 160
specifying transformations, 159
starting the CDT, 153
stopping the CDT, 153
transforming elements, 162
troubleshooting
abandoned records errors, 173
cache refresh errors, 172
FK check errors, 171
rollback scripts, 174
unexpected errors, 173
user interface
Comparison Results window, 157
Deployment Explorer window, 156
Status panel, 158
configuring
WebLogic, 119
configuring application server, 38
JDK upgrades, 39
copying WMS label formats, 63
copying WMS node
rapid deployment tool, 104
creating EAR
for WebLogic, 123
for WebSphere, 129
cross-site script vulnerabilities
malicious HTML, 83
prevention, 83
enabling html encoding, 83
custom classes deployment, 109

D

data migrator, 99
connection settings, 86
log files, 100
errors, warnings, 100
modifying migrator.properties, 85
properties file, 85
restart file, 100
task definition files, 99
database
connection settings, 84
modifying yfs.properties, 84
override properties, 85
database security, 20
credit card encryption, 20
database sizing, 50
capacity planning, 50
disk estimation
DOM module, 51
nWMS module, 54
future disk estimation, 59
database user privileges, 43
DB2, 47
Oracle, 43
oracle
administrative user, 43
application user, 44
SQL server, 49
database verification, 97
db-extn-sqlserver, 124, 131
dbverify script, 97
DB2
codepage selection, 47
configuring, 47
copying database driver, 47
installation, 47

DB2 database
production environment, 47
running scripts, 48
running customDB views, 48
sizing, See Also database sizing, 44, 47
deploying
custom classes, 109
deploying configuration data, 139
deploying EAR
on WebLogic, 123
on WebSphere, 125
deploying Yantra 7x
on WebLogic, 115
on WebSphere, 125
deploying Yantra7x, 105
deployment architecture
analyzing
authentication mechanism, 13
current security infrastructure, 13
data encryption, 14
network topology, 14
disk estimation
DOM module, 51
before estimating, 51, 52
estimation methodology, 51
steps involved, 53
future requirements, 59
nWMS and DOM module
steps involved, 56
nWMS module, 54
before estimating, 54, 55
estimation methodology, 54
nWMS module only
steps involved, 58
DOM module
disk space estimation, 53
DOM module disk estimation, 51
driver entities, 141

E

EAR (Enterprise Archive), 123
EJB module
adding resource references, 132
EJB (Enterprise JavaBeans), 16
ejbdeploy.sh, 136
encrypting properties, 82
enterprise javabeans. See EJB, 16
errors, 46
estimates of database, 50

F

factory defaults, 95
installer restart file, 96
modfiying loadDefaults file, 95
modifying migrator.properties, 95
specifying US geography defintions, 96
specifying US region defintions, 96
file separator, 28, 82

G

generating EJB stubs, 135
Graph display on WebLogic, 120
Graph display on WebSphere, 128

H

history database, 51, 54
HP-UX
ncurses installation, 73
HTML characters
impacting display, 83
HTTP
in-memory session replication, 125
HTTP (hypertext transfer protocol), 16
HTTPS
enforcing secure protrocol, 110
HTTPS (hypertext transfer protocol secure), 17
hypertext transfer protocol secure. See HTTPS, 17
hypertext transfer protocol. See HTTP, 16

I

installation checklist, 1
installing
DB2
Oracle

SQL Server
installing analytics, 26
installing application server, 38
installing application tier software, 37
installing carrier server, 26
installing database software, 41
DB2. See DB2
Oracle. See Oracle
SQL Server. See SQL Server
installing DB2, 46
installing jasperreports, 63
installing mobile application, 67
installing Oracle, 42
installing print server, 61
installing SQL Server, 48
installing weighing scale, 65
installing Yantra 7x, 25
on remote machine, 28
setting permissions, 28
on UNIX and LINUX, 26
on Windows, 27
UNIX criteria, 26
installing Yantra 7x language pack, 31
internet access
running Yantra 7x configurator, 82
Internet Explorer security
adding trusted website, 23
browser settings, 21
intgeration server setting
modifying configurator UI, 91
modifying management.properties, 90
modifying yfs.properties, 90
invoking
custom classes, 109

J

JAAS (Java Authentication and Authorization Service), 13
jasperreports
installation, 63
Java Authentication and Authorization Service. See JAAS, 13
java messaging service. See JMS, 16
java naming and directory interface. See JNDI, 17
java plugin
changing yfs.properties file, 82
clearing cache, 137
java protocol
disabling, 16
EJB, 16
HTTP, 16
JMS, 16
java protocols
securing, 16
COM+, 18
EJB, 17
HTTP API tester, 18
JavaServer pages. See JSP, 17
jcert.jar, 118
jdbcurl, 84, 86
JMS (java messaging service), 16
JNDI clean up
stale entries, 118
JNDI registry, 119
clean up, 118
stale entries, 118
JNDI registry clean up, 118
JNDI (java naming and directory interface), 17
jnet.jar, 118
JSP (JavaServer Pages), 17
jsse.jar, 118
JVM settings
classpath, 127
custom properties, 127
for WebSphere, 126
WebSphere properties, 126

L

LAN (local area network), 15
language pack
creating EAR, 35
creating ear, 35
deploying EAR, 35
installing, 32
on remote computer, 33
on UNIX and LINUX, 32
on Windows, 32
loading factory defaults, 33

loading language pack translations, 34
setting up properties, 33
switching base language, 34
LDAP user authentication, 86
assumption, 86
for application consoles, 86
modifying configurator UI, 88
modifying weblogic startup file, 88
modifying yfs.properties, 87
password expiration, 86
change password link, 87
expire in days, 86
properties, 87
LDAP (lightweight directory access protocol), 13
license agreeement, 177
ligtweight directory access protocol. See LDAP, 13
LLM (loftware label manager), 61
lloftware label manager
modifying yfs.properties, 62
loading factory defaults,database factory defaults, 95
local area network. See LAN, 15
loftware label manager, 61
bar code label, 62
defining printers, 63
designing label, 62
for UNIX systems, 63
supported modes, 62
log files
best practices, security, 18
for data migrator, 100
setting up, 88
logging
modifying yfs.properties, 89
properties, 88
utility, 88
login.jsp, 112
log4j configuration file, 89
log4j utility, 88
LPM (loftware print server), 61

M

maintaining accurate entries, 119
maximum performance, 4
Microsoft .NET, 72
mixed protocol
filter initialization parameter
HTTPListenPort, 112
HTTPSListenPort, 112
SecuredQueryStringPatterns, 112
SecuredURLPatterns, 112
UnchangedQueryStringPatterns, 112
UnsecuredQueryStringPatterns, 112
UnsecuredURLPatterns, 112
mixed protocols, 83, 110
configuring filter, 110
enforcing HTTPS, 110
filter mechanics, 113
secure,unsecure, 110
securing login information, 114
mobile application
Activesync for copying files, 69
installation, 67
mobile installation
cold boot reinstallation, 71
installing libiconv
HP-UX Itanium, 73
installing ncurses, 73
steps to install, 68
mobile terminals
device requirements
barcode scanner, 8
keys, 8
minimum memory, 9
network connectivity, 9
screen resolution, 9
synchronization, 8

N

named pipes, 48
namedwebservices.xml, 120
ncurses, 73
network topology
deployment over internet, 14
deployment over LAN, 15
deployment over VPN, 15
networked WMS module disk estimation, 54
non-Yantra files, 90

nWMS and DOM module
disk space estimation, 56
nWMS module only
disk space estimation, 58

O

operating system permissions, 19
Oracle
configuring, 44
copying JDBC driver, 43
installation, 42
running create instance, 42
Oracle database
production environment, 44
creating views, 44
running scripts, 45
setting connection properties, 84
query rewrite priveleges, 45

P

password expiration, 86
Pocket PC mobile terminal
reintalling on cold boot, 71
PocketPC mobile terminal
installing Yantra, 67
post installation recommendations, 18
pre-compile JSPs
WebLogic utility, 115
pre-compiling JSP files, 106
Preparing the Database Server, 42
print server
configuration utility, 62
installing loftware components, 62
print server installation, 61
procuring storage, 52, 55
product licensing, 177
software agreement, 177
third-party tools, 186
product packaging, 177
production environment, 141
properties file
for data migrator, 85
properties, encrypting, 82

R

RDT (rapid deployment tool), 104
recommended data encryption
SSL 128-bit, 14
VPN 3DES, 14
RegionSchema-US.sql, 96
remote method invocation. See RMI, 17
restart file
for data migrator, 100
restart files
preserving, 100
retention policy, 52
reverse proxy, 112
RMI (remote method invocation), 17
rollback of configuration data, 174
running
DB2 scripts, 48
Oracle scripts, 46
SQL Server scripts, 49

S

secure protocol, 110
secure socket layer. See SSL, 14
security
database security, 20
deployment architecture, 12
java protocol See java protocol
web security See web security
security plan
creating, 11
secutiry
Internet Explorer settings, 21
session replication, 125
idempotent flag, 125
including session descriptors, 125
set up
utility script files, 94
UNIX permissions, 94
seting up
properties file, 82
Setting Oracle Database Connection Properties, 84
setting up
Agent server, 90

database verification tool, 97
Integration server, 90
Oracle scripts, 45
WebLogic scripts, 116
webservices, 120
setting up Yantra 7x, 79
setting up Yantra7x
properties file, 80
SQL Server
configuring, 49
SQL Server database
case-insensitive mode, 49
collation property, 49
copying database driver, 49
enabling named pipes, 48
enabling TCP/IP, 48
estimating character set, 49
production environment, 49
running scripts, 49
sizing, See Also database sizing, 49
SSL accelarator, 14
SSL (secure socket layer), 14
stack traces
eliminating, 120
staging environment, 141
stale entries, 118
eliminating, 118
statistics monitoring, 137
supressing password prompt, 114
System Console, 129
system requirements
minimum, 4

T

task defintion file
childtask element, 100
task element, 99
taskinfo element, 100
task defintion files
for data migrator, 99
TCP/IP, 48
third-party licensing, 186
time-triggered transactions
override agent critetria, 102
transactional database, 51, 54
types of order, 52

U

unsecure protocol, 110
US_ZipcodeLocation.sql, 96

V

verifying database, 97
extra objects, 98
verifying deployment, 137
virtual host configuration, 128
virtual private network. See VPN, 15
VPN (virtual private network), 15
VT220 Mobile Terminal, 67, 73
installing, 73
steps to install, 74

W

WAR file
adding webservices elements, 122
WAR module
adding resource references, 133
web security
routing, 19
session security, 19
web server executables, 20
writing log files, 18
WebLogic
configuring for Yantra 7x, 119
creating EAR
configuring environment variables, 123
exporting ANT_OPTS, 123, 131
extended database, 124
file descriptors, 123
steps involved, 123
yantra.ear, 124
creating EAR, deploying EAR, 123
eliminating stack traces, 120
preparing for JNDI clean up, 118
running webservices, 124

script properties, 116
setting client character display, 136
setting HTTP in-memory session replication, 125
using webservices, 120
webservices properties, 121
WebLogic application server
setting up, 116
weblogic.jar, 119
weblogic.xml, 125
webservices, 120
on WebLogic, 124
on WebSphere, 129
properties on WebLogic, 121
properties on WebSphere, 129
WebSphere
application clients
invoking Yantra 7x EJBs, 135
avoiding warning messages, 126
configuring EAR, 132
adding resource references, 132, 133
exporting EAR, 134
importing the EAR, 132
configuring JVM settings, 126
configuring virtual host, 128
creating EAR, 129
extended database, 131
setting environment variables, 130
steps involved, 130
deploying EAR, 125, 134
JVM properties, 126
making EJB calls, 135
preparing for deployment, 126
running webservices, 129
specifying memory parameters, 126
webservices properties, 129
web.xml, 110
weighing scale
installation, 65
steps to install, 66
WinCE mobile terminal, 67
installing Yantra, 67
WinCE mobile terminals
reinstalling on cold boot, 72
WMS installation
defining printers, 63

Y

Yantra 7x
deploying, 105
Yantra 7x Applications, 179
Yantra 7x components, 178
application consoles, 178
configurator, 178
platform, 178
Yantra 7x Configurator
properties, 82
Yantra 7x Delivery and Service Scheduling, 183
Yantra 7x Distributed Order Management, 180
Yantra 7x Inventory Synchronization, 182
Yantra 7x Logistics Management, 181
Yantra 7x Mobile Application
installation
PocketPC mobile terminal, 67
Yantra 7x Platform, 179
Yantra 7x Reverse Logistics, 182
Yantra 7x Supply Collaboration, 180
Yantra 7x Utilities
development
configuration deployment tool, 101
Yantra 7x utilities
development, 101
installation, 94
dbverify, 95
loadDefaults, 95
production, 104
getServerDetails, 104
runtime, 101
agent server, 102
agent trigger, 102
Integration server, 101
setting up classpath, 102
upgrade, 98
MigrationValidator, 98
migrator, 98
Yantra 7x Warehouse Management System, 183
yantra.ear, 124
yantraejb.jar, 107
yantra.jar, 106, 136

yantra.war, 107
yantrawebservices.war, 107
yantra_ejbstubs.jar, 136
yantra_indexadds.sql, 98
yantra_indexdrops.sql, 98
yantra_sequence.sql, 97
yantra_tablechanges.sql, 98
yantra_tabledrops.sql, 98
ycpwsbe.jar, 121
yfs.properties
  Configuration Deployment Tool, 151
  dblogin.jdbcurl, 84, 86
yfssqlserver_master_db_script.cmd, 50
yfs_master_db_script.log, 46
yfs_master_db_script.sql, 45, 46
yfs_seq_db2.sql, 48
yfs_tables.sql, 45, 48