



System Management User Guide

Release 7.5 SP1

May 2006



Copyright Notice

System Management User Guide, Release 7.5 SP1

Copyright © 2004 - 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This documentation, the software it describes, and the information and know-how they contain constitute the unpublished, proprietary, confidential and valuable trade secret information of Yantra Corporation, its affiliated companies or its or their licensors, and may not be used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 © 2004 -
2006

Contents

Preface

Intended Audience	vii
Structure	vii
Yantra 7x Documentation	vii
Conventions	ix

1 Introduction

1.1 Understanding Yantra 7x System Components	1
1.1.1 Application Hosts	1
1.1.1.1 Hosts.....	2
1.1.1.2 Application Servers.....	2
1.1.2 Clients	2
1.1.3 Application Programming Interface (APIs).....	2
1.1.3.1 User Exits	3
1.1.3.2 Services (Custom APIs).....	3
1.1.4 Agents and Integration Servers.....	3
1.1.4.1 Time-Triggered Transactions and Agent Servers	3
1.1.4.2 Integration Servers	4
1.1.5 JMS Queues	4
1.1.6 Database	4

2 System Management Tasks

2.1 Monitoring Yantra 7x	9
2.1.1 Running the Health Monitor Agent	9

2.1.2	Monitoring Application Hosts	9
2.1.2.1	Server Name	10
2.1.2.2	Server Heartbeat.....	10
2.1.2.3	Application Server Response Time	10
2.1.3	Monitoring APIs.....	11
2.1.3.1	API Response Time.....	11
2.1.3.2	Other API Statistics	12
2.1.3.3	User Exit Statistics	13
2.1.4	Monitoring Agents and Integration Servers	13
2.1.4.1	Health Monitor Agent	13
2.1.4.2	Server Heartbeat.....	14
2.1.4.3	Agent Pending Tasks.....	15
2.1.4.4	Other Agent Statistics	15
2.1.4.5	Integration Server Statistics	16
2.1.5	Monitoring JMS Queues	16
2.1.5.1	Number of Messages in a JMS Queue	16
2.1.6	Data Published for Health Monitor Alerts	17
2.1.7	Monitor Groups	18
2.2	Accessing the System Management Console.....	19
2.3	Navigating in the System Management Console.....	19
2.3.1	Detecting a Problem	20
2.3.2	Opening Multiple Monitor Screens at the Same Time	20
2.4	Restricting Access to the System Management Actions	20
2.5	Resolving Problems with Application Server Response Time.....	21
2.5.1	Application Server Cache Level.....	21
2.5.2	Client Load Too High.....	22
2.5.3	JSPs Not Precompiled.....	22
2.5.4	Frequent JVM Full Garbage Collection	23
2.6	Resolving Application Server Down Alerts	23
2.7	Resolving Problems with API Response Time	23
2.7.1	Slow Response Time for a User Exit	23
2.7.2	Slow Response Time for one JVM.....	24
2.8	Resolving Problems with Agent Server Pending Tasks.....	25
2.8.1	Agent Server is not Running	25
2.8.2	Agent Server Needs More Threads	25

2.8.3	Errors are Being Generated	25
2.9	Resolving Problems with JMS Queue Number of Messages	26
2.9.1	Message Consumer Process is not Running	26
2.9.2	Processing Rate is Low for Message Consumer Process	26
2.10	Tracing a Component.....	27
2.11	Enabling an Agent Trace	27
2.12	Stopping a Component Trace	27
2.13	Viewing Application Server Cache Information.....	28
2.14	Clearing Database Cache for an Application Server	28
2.15	Disabling Cache or Changing Database Cache Limits for an Application Server	29
2.16	Viewing Application Server Properties	29
2.17	Shutdown, Suspend, or Resume an Agent or Integration Server Instance ..	30
2.18	Starting an Agent Server.....	30
2.19	Starting an Integration Server	31
2.20	Scheduling an Agent.....	32
2.21	Increasing Threads on an Agent or Integration Server Instance.....	32
2.22	Viewing Properties for an Agent or Integration Server Instance	33
2.23	Changing Monitor Groupings.....	33
2.24	Setting Notification of an Application Server Shutdown.....	34
2.25	Setting Notification of an Agent or Integration Server Unexpected Termination.....	34
2.26	Setting Notification of Threshold Reached or Exceeded	34
2.27	Starting the Health Monitor Agent	34

3 System Management Console Screens

3.1	System Management Console	39
3.2	Application Hosts Summary a	47
3.3	Application Server Details	48
3.4	API Groups Summary	50
3.5	API Details	50
3.6	Agent and Integration Server Summary	52
3.7	Agent and Integration Server Detail	53
3.8	Agent and Integration Server Instance Detail	57
3.9	JMS Queue Summary	59

3.10	JMS Queue Details.....	60
3.11	Monitor Group Details	61
3.12	Traced Components List	65
3.13	Trace Component Details	65
3.14	Table Level Cache List	67
3.15	Live Agent or Integration Servers	69

Index

Intended Audience

This manual provides system management information for individuals responsible for maintaining and monitoring the Yantra 7x System Management application.

Structure

This manual contains the following sections:

Chapter 1, "Introduction"

This chapter provides a high-level introduction to the components that make up Yantra 7x.

Chapter 2, "System Management Tasks"

This chapter explains how to perform specific tasks in the Yantra 7x System Management Console.

Chapter 3, "System Management Console Screens"

This chapter explains the field level descriptions of the Yantra 7x System Management console user screens.

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

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.
/ 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.

Introduction

The *Yantra 7x System Management Guide* contains information relevant to managing and monitoring the Yantra 7x application.

This chapter provides a high-level introduction to the components that make up Yantra 7x.

1.1 Understanding Yantra 7x System Components

Yantra 7x is comprised of many different components. On a day-to-day basis the following components need to be monitored to ensure Yantra 7x is running properly and efficiently:

- [Application Hosts](#)
- [Clients](#)
- [Application Programming Interface \(APIs\)](#)
- [Agents and Integration Servers](#)
- [JMS Queues](#)
- [Database](#)

1.1.1 Application Hosts

Application hosts are the physical machines on which one or more Yantra 7x application server processes are run. The application servers must constantly be running to give all client users access to the Yantra 7x Application Consoles and Configurator.

1.1.1.1 Hosts

The physical machine that runs the application server(s) is known as the host.

1.1.1.2 Application Servers

The application servers are the processes that handle requests from various places to provide the real-time access to the features and application logic within the Yantra 7x application. The most common type of requests that an application server handles are the requests originating from clients using the Yantra 7x Application Consoles. The applications servers are always deployed using an industrial-strength server application such as BEA Weblogic.

It is important to ensure that the application servers are constantly running and are performing well to ensure all requests are being fulfilled quickly. If the servers are not running properly, the application slows down, and the users of the Yantra 7x Application Consoles most likely will find it very difficult to perform their tasks. Multiple application server processes can be run (on the same or different host machines) if the volume of requests is high. It is important to run the right number of application servers based on the volume of requests in your installation.

1.1.2 Clients

Clients are processes that connect to the application servers to fulfill a request. Most commonly, clients are connections originating from the users accessing the Yantra 7x Application Consoles.

1.1.3 Application Programming Interface (APIs)

APIs are integration points that provide access to the rich business logic and features available in Yantra 7x to external systems. APIs are used extensively in Yantra 7x for operations such as creating an order, adjusting inventory levels, and retrieving details of a shipment. The Yantra 7x Application Consoles also call APIs to retrieve data and perform updates. See the *Yantra 7x Customization Guide* for more information about APIs.

1.1.3.1 User Exits

Some APIs call user exits. User exits allow you to extend or override key business algorithms with Yantra 7x. For more information about user exits, see the *Yantra 7x Customization Guide*.

1.1.3.2 Services (Custom APIs)

The Service Definition Framework allows you to create custom APIs that are called through integration points or through the Yantra 7x Application Consoles. For more information about the Service Definition Framework see the *Yantra 7x Platform Configuration Guide*.

1.1.4 Agents and Integration Servers

Agents and integration servers are processes that run in the background to perform various tasks.

1.1.4.1 Time-Triggered Transactions and Agent Servers

A time-triggered transaction is a program that performs a variety of individual functions, automatically and at specific time intervals. It is not triggered by conditions, events, or user input. There are three types of time-triggered transactions:

- * Business process transactions - responsible for processing day-to-day transactions.
- * Monitors - watch and send alerts for processing delays and exceptions.
- * Purges - clear out data that may be discarded after having been processed.

For information on using the time-triggered transactions provided by Yantra 7x, see the *Yantra 7x Platform Configuration Guide* and the *Yantra 7x Customization Guide*.

The process that runs the time-triggered transactions is known as an agent server. Agents pick up the appropriate "pending tasks" for the agent and process them one at a time.

An agent server can run multiple time-triggered transactions. Additionally, each time-triggered transaction can run with one or more threads. It is also possible to run multiple instances of the same agent server.

The correct configuration depends on the volume of transactions in your system.

1.1.4.2 Integration Servers

An Integration Server is a process that manages asynchronous services, such as messages to and from external systems.

Integration servers allow Yantra 7x to collaborate with different systems, organizations, and businesses—all through a standard, uniform interface to all systems.

Integration Servers are configured through the means of the Service Definition Framework. For more information on the Service Definition Framework see the *Yantra 7x Platform Configuration Guide*.

An integration service definition contains one or more sub-services each of which are their own asynchronous service. Each of these sub-services can be run with one or more threads. When an integration server is launched, it will process the messages for the asynchronous component in the service definition. Additionally, multiple instances of an integration server can be run at the same time.

1.1.5 JMS Queues

External message queueing software can be used for the Yantra 7x application to communicate with external systems. Yantra 7x supports the following JMS message queue software:

- BEA Weblogic JMS
- IBM Websphere MQ

JMS queues can be configured and used within Yantra 7x's Service Definition Framework. For more information on integrating with JMS systems see the *Yantra 7x Integration Guide*.

1.1.6 Database

The Yantra 7x application always works with a single database instance. A single database instance allows the various components of Yantra 7x to work together seamlessly. It is extremely critical to ensure the single database instance is working properly at all times. There are many powerful tools in existence that can help you monitor your database,

therefore, Yantra 7x does not provide any additional database administration or monitoring tools as part of the Yantra 7x System Management.

Part I

System Management Tasks

The chapters in this section provide an in-depth explanation of the various tasks that can be performed in the System Management Console screens.

This section contains the following chapters:

- [Monitoring Yantra 7x](#)
- [Accessing the System Management Console](#)
- [Navigating in the System Management Console](#)
- [Restricting Access to the System Management Actions](#)
- [Resolving Problems with Application Server Response Time](#)
- [Resolving Application Server Down Alerts](#)
- [Resolving Problems with API Response Time](#)
- [Resolving Problems with Agent Server Pending Tasks](#)
- [Resolving Problems with JMS Queue Number of Messages](#)
- [Tracing a Component](#)
- [Stopping a Component Trace](#)
- [Viewing Application Server Cache Information](#)
- [Clearing Database Cache for an Application Server](#)
- [Disabling Cache or Changing Database Cache Limits for an Application Server](#)
- [Viewing Application Server Properties](#)

- Shutdown, Suspend, or Resume an Agent or Integration Server Instance
- Starting an Agent Server
- Starting an Integration Server
- Scheduling an Agent
- Increasing Threads on an Agent or Integration Server Instance
- Viewing Properties for an Agent or Integration Server Instance
- Changing Monitor Groupings
- Setting Notification of an Application Server Shutdown
- Setting Notification of an Agent or Integration Server Unexpected Termination
- Setting Notification of Threshold Reached or Exceeded
- Starting the Health Monitor Agent

System Management Tasks

This chapter explains how to perform specific tasks in the Yantra 7x System Management Console.

2.1 Monitoring Yantra 7x

Yantra 7x System Management allows you to monitor the application in real-time so that problems can be acted upon immediately. This section explains the basics of monitoring and how to get started.

Various statistics about each of the system components are collected behind the scenes while the application is running. These statistics are persisted into the Yantra 7x database at intervals of 10 minutes. This is referred to as the "persist interval" through-out this document.

2.1.1 Running the Health Monitor Agent

The Health Monitor Agent executes the appropriate configured services whenever an alert condition occurs (as explained below). To run the Health Monitor Agent run the `startHealthMonitor.sh` script file located in your `YFS_HOME/bin` directory.

If using Weblogic JMS or MQSeries queues, ensure the classpath includes the Weblogic or MQSeries specific jars.

2.1.2 Monitoring Application Hosts

Application hosts must be constantly running to ensure high availability of the application. Additionally, the response time of servers should be monitored to ensure users are getting fast responses from the Yantra 7x Application Consoles. Yantra 7x System Management enables you to easily monitor the status and response times of your application hosts.

2.1.2.1 Server Name

To monitor your application servers, you must give each server a unique name so that Yantra 7x System Management can uniquely identify each server. To give a unique name to an application server, add the following command line parameter to the java command within the server start up script:

```
-DYantra.Server.Name=<unique name>
```

Even if multiple server processes are run on the same machine, it is mandatory to give each server process a unique name. Therefore, it may be necessary to create separate start up scripts even when starting multiple instances of the server on the same machine.

2.1.2.2 Server Heartbeat

Yantra 7x System Management tracks the status of the application servers by recording the server "heartbeat" while the server is running. If the server goes down, the heartbeat stops getting recorded. If a server with the same unique ID is brought back up, the heartbeat resumes. For more information on purging heartbeat records see [Section 2.1.4.1, "Health Monitor Agent"](#).

2.1.2.2.1 Alert when Server Goes Down

It is possible to configure a service to be executed whenever an application server goes down (intentionally or unexpectedly). This service can do anything at all including sending an e-mail message to a system administrator or creating an alert in a system administrator's inbox. For information on specifying the service to execute when an application server goes down, see the information about configuring the Health Monitor Rules in the *Yantra 7x Platform Configuration Guide*. For more information about the data available for the service, see [Section 2.1.6, "Data Published for Health Monitor Alerts"](#).

2.1.2.3 Application Server Response Time

The response time of each HTML request is calculated for each application server. During every persist interval, the minimum, maximum, and average of all the response times is recorded.

2.1.2.3.1 Alert when the Response Time Threshold is Exceeded

It is possible to configure a service to be executed if the average response time of an application server goes above a threshold limit for three consecutive persist intervals. This service can do anything at all including sending an e-mail message or creating an alert for a system administrator. For information on specifying the service to execute when the threshold is exceeded, see the information about configuring the health monitor rules in the *Yantra 7x Platform Configuration Guide*. For more information about the data available for the service, see [Section 2.1.6, "Data Published for Health Monitor Alerts"](#).

2.1.2.3.2 Default Value for Application Server Response Time Threshold

The default threshold value for application server response time is set using the `yantra.hm.appserver.threshold` property in the `yfs.properties` file. Change this property to set the default threshold for all application servers.

2.1.2.3.3 Changing the Threshold for a Specific Application Server

A specific threshold can be set for specific application servers as well. For information on changing the threshold of a specific application server see [Section 3.3, "Application Server Details"](#).

2.1.3 Monitoring APIs

To ensure that the through-put of the application is at optimal levels, the APIs must be executing with good response times. API response times also can effect the Yantra 7x Application Consoles users because the Application Consoles fetch and update all information using APIs.

2.1.3.1 API Response Time

The response time is calculated every time an API is called. During every persist interval, the minimum, maximum, and average response time of all the calls is recorded.

2.1.3.1.1 Alert when the Response Time Threshold is Exceeded

It is possible to configure a service to be executed if the average response time of an API goes above a threshold limit for three consecutive persist intervals. This service can do anything at all including

sending an e-mail message or creating an alert for a system administrator. For information on specifying the service to execute when the threshold is exceeded, see the information about configuring the health monitor rules in the *Yantra 7x Platform Configuration Guide*. For more information about the data available for the service, see [Section 2.1.6, "Data Published for Health Monitor Alerts"](#) on page 2-17.

2.1.3.1.2 Default Value for API Response Time Threshold

The default threshold value for API response time is set using the `yantra.hm.api.threshold` property in the `yfs.properties` file. Change this property to set the default threshold for all APIs.

2.1.3.1.3 Changing the Threshold for a Specific API

There are a large number of APIs within Yantra 7x that can do a great number of different things. Therefore, it is likely that finer control over the API response time thresholds is required. It is possible to set a specific threshold value for a specific API as well. For more information on setting a specific API threshold see [Section 3.5, "API Details"](#).

Note: Setting the appropriate API thresholds is not an exact science. Experimentation is required to find the correct threshold for each API to ensure that alerts are only sent when absolutely necessary. Keep in mind that some APIs may execute with variable sizes of data that have a direct correlation to the response time of that API. For example, the `createOrder` API should have a much larger response time for an order with 100 order lines compared to a `createOrder` API call for an order with 1 order line.

2.1.3.2 Other API Statistics

Yantra 7x System Management also records the number of invocations for each API that has happened during each persist interval.

Additionally, some of the most important APIs record statistics that are specific to that API. For example, the `createOrder` API records the number of orders created and number of order lines created during each persist interval.

2.1.3.3 User Exit Statistics

If an API has user exits that have been implemented, then statistics for each user exit call are recorded at each persist interval. The statistics collected for each user exit call are minimum, maximum, and average response time as well as the number of invocations.

It is not possible to set a threshold or configure a service to be executed based on the response time of a user exit. However, the calculated API response time is inclusive of the user exits called within it. Therefore, if a user exit suddenly starts to respond slowly, the API response time also increases.

2.1.4 Monitoring Agents and Integration Servers

To maximize the throughput of the application, the agents should be monitored to ensure that they are able to process all of the pending tasks within an acceptable time frame. If an agent is not able to process its tasks fast enough, the pending jobs accumulate and cause a bottleneck in the system.

2.1.4.1 Health Monitor Agent

The health monitor appears as a normal agent and provides the following abilities:

- ability to shutdown the entire health monitor agent
- allowing cache changes
- viewing server properties
- changing logging parameters and
- sub-service visibility.

A few of the statistics provided by health monitor agents are:

- Processing Rate
- Number of heartbeats purged
- Number of snapshots purged
- Application server down alerts
- Server unavailable alerts
- Threshold reached alerts

- Heartbeats monitored

The `YFS_SNAPSHOT` table stores the statistical details of pending tasks of transactions collected by the agent servers. The parameter `CollectPendingJobs` in time-triggered agents controls whether records are inserted in the table. The health monitor deletes the records from this table after the default purge interval of 30 days.

The heartbeat records in the `YFS_HEARTBEAT` table are also purged by the health monitor agent with a default purge interval of 30 days.

The health monitor schedules a purge once every 24 hours to purge the snapshot and heartbeat records that are older than 30 days. You can modify this purge interval from 30 days to suit your needs by entering the `yantra.hm.purge.interval` parameter in the `yfs.properties` file.

2.1.4.2 Server Heartbeat

Yantra 7x System Management tracks the status of the agent and integration servers by recording the server "heartbeat" while the server is running. If the server goes down, the heartbeat stops getting recorded. If a server with the same name is brought back up, the heartbeat resumes. For more information on purging heartbeat records see [Section 2.1.4.1, "Health Monitor Agent"](#).

2.1.4.2.1 Alert when Agent or Integration Server Terminates Unexpectedly

It is possible to configure a service to be executed whenever an agent or integration server goes down unexpectedly. This service can do anything at all including sending an e-mail message to a system administrator or creating an alert in a system administrator's inbox. For information on specifying the service to execute when an agent or integration server goes down, see the information about configuring the health monitor rules in the *Yantra 7x Platform Configuration Guide*. For more information about the data available for the service, see [Section 2.1.6, "Data Published for Health Monitor Alerts"](#).

Shutting down an agent or integration server through the System Management Console (or pressing `Cntrl+C` on the command line window) will not generate an alert.

2.1.4.3 Agent Pending Tasks

The number of pending tasks of every agent is recorded during every persist interval, unless the `CollectPendingJobs` criteria parameter for the agent is set to `N` in the Agent Criteria Details. For more information on configuring agents, see the *Yantra 7x Platform Configuration Guide*.

2.1.4.3.1 Alert when the Pending Tasks Threshold is Exceeded

It is possible to configure a service to be executed if the number of pending tasks for an agent goes above a threshold limit. This service can do anything at all including sending an e-mail message or creating an alert for a system administrator. For information on specifying the service to execute when the threshold is exceeded, see the information about configuring the health monitor rules in the *Yantra 7x Platform Configuration Guide*. For more information about the data available for the service, see [Section 2.1.6, "Data Published for Health Monitor Alerts"](#).

2.1.4.3.2 Default Value for Agent Pending Tasks Threshold

The default threshold value for agent pending tasks is set using the `yantra.hm.agent.threshold` property in the `yfs.properties` file. Change this property to set the default threshold for all agents.

2.1.4.3.3 Changing the Pending Tasks Threshold for a Specific Agent

There are a large number of agents within Yantra 7x that can do a great number of different things. Therefore, it is likely that finer control over the agent pending tasks thresholds is required. It is possible to set a specific threshold value for a specific agent as well. For more information on setting a specific agent threshold see [Section 2.1.4.3.3, "Changing the Pending Tasks Threshold for a Specific Agent"](#).

2.1.4.4 Other Agent Statistics

Yantra 7x System Management also records the processing rate for each agent during each persist interval.

Additionally, some of the most important agents record statistics that are specific to that agent. For example, the schedule order agent records the number of orders scheduled and number of orders backordered during each persist interval.

For more information on other agent statistics see the time-triggered transaction reference in the *Yantra 7x Platform Configuration Guide*.

2.1.4.5 Integration Server Statistics

Yantra 7x System Management records the processing rate as well as the minimum, maximum, and average response times for integration servers for each persist interval.

It is not possible to set a threshold or configure a service to be executed for any of the statistics collected for integration servers.

2.1.5 Monitoring JMS Queues

To ensure that the integration points between Yantra 7x and other systems that are using JMS messaging queues are working properly, the number of messages in the queues should be monitored. If the messages in the queue are not being processed quickly enough or not being processed at all, then something has gone wrong at the integration point. If the problem at this integration point is not resolved quickly, then more problems are sure to happen somewhere further along in the process.

2.1.5.1 Number of Messages in a JMS Queue

The number of messages for the JMS queues being monitored are recorded during every persist interval. Note that this statistic is only recorded for JMS queues that are actually being monitored. For information on how to monitor a JMS queue refer to [Section 3.11, "Monitor Group Details"](#).

Note: The JMS queue monitor retrieves data every 10 minutes. For information about monitoring more real-time data for your JMS queues, see the documentation provided by your JMS server software vendor.

2.1.5.1.1 Alert when the Number of Messages Threshold is Exceeded

It is possible to configure a service to be executed if the number of messages in a JMS queue goes above a threshold limit. This service can do anything at all including sending an e-mail message or creating an alert for a system administrator. For information on specifying the service

to execute when the threshold is exceeded, see the information about configuring the health monitor rules in the *Yantra 7x Platform Configuration Guide*. For more information about the data available for the service, see [Section 2.1.6, "Data Published for Health Monitor Alerts"](#).

There is no default value for the number of messages threshold limit. This threshold must be specified when adding a JMS queue to monitor. For more information see [Section 3.11, "Monitor Group Details"](#).

2.1.6 Data Published for Health Monitor Alerts

When alerts are generated within the health monitor agent (for threshold exceeded, application server down, agent or integration server unexpected termination), the data for the alert is published in XML format.

For threshold exceeded alerts, the XML format is:

```
<Alert ServiceName="" ServiceType="" Threshold="" Value="" Units=""/>
```

This XML contains the following attributes:

Table 2–1 Threshold Exceeded Alert Attributes

Attribute	Description
ServiceName	The name of the component for which the threshold has been exceeded.
ServiceType	The type of the component for which the threshold has been exceeded. This will take one of the following values: <ul style="list-style-type: none"> • APPSERVER • API • AGENT • JMS
Threshold	The current threshold limit set for the component.

Table 2–1 Threshold Exceeded Alert Attributes

Attribute	Description
Value	The current value of the statistic that exceeds the threshold value.
Units	The unit of measure in which the value and threshold amounts are returned in. This will take one of the following values: <ul style="list-style-type: none"> • Milliseconds • PendingTasks • Messages

For application server down, agent or integration server unexpected termination alerts, the XML format is:

```
<Alert ServerName="" HostName="" ServerStartTime="" />
```

This XML contains the following attributes:

Table 2–2 Application Server Down & Agent or Integration Server Termination Alert Attributes

Attribute	Description
ServerName	The name of the server that caused the alert.
HostName	The application host name on which the server that caused the alert was running.
ServerStartTime	The time the server that caused the alert was started.

2.1.7 Monitor Groups

The Yantra 7x System Management console gives visibility to all of the statistics mentioned in this document. Each application server, API, agent, integration server, and JMS queue can be viewed and monitored within the console.

Because there are so many different components to monitor, they can be organized into meaningful monitor groups. Yantra 7x provides default groupings based on similar business logic. For example, all order related

APIs and agents are grouped into the "Order" monitor group. These default groupings can be changed to accommodate any other desired organization. For example, you might choose to group all heavily used agents together. For more information on monitor groups see [Section 3.1, "System Management Console"](#).

2.2 Accessing the System Management Console

To access the Yantra 7x System Management console:

1. Point your browser to:

```
http://<Server where Yantra 7x is installed>/yantra/console/start.jsp
```

The Sign In window displays.

2. Enter your login ID and password and click on the Sign In button. The Yantra 7x Application Consoles Home Page is displayed.
3. From the menu bar, choose System > System Management Console.

Note: Yantra recommends you increase the memory available to the java plugin for every computer that runs the System Management console. To increase the memory available, open the java plugin settings from the windows control panel. Specify the following java runtime parameters: `-Xms128M -Xmx128m`.

Note: If both the Yantra 7x Configurator and the Yantra 7x System Management are opened at the same time, and if a dialogue window is opened in either application, the other will stop responding to user input until that dialogue window is closed. This is due to a bug in the Java platform.

2.3 Navigating in the System Management Console

The screens within the System Management console display the current state of various components within the entire Yantra 7x application. There are three main types of screens:

- Main View of the Yantra 7x application - The main screen that always remains open and displays the overall status of the Yantra 7x application.
- Monitor Group Summary Tree and Graphs - The second level of screens that display the overview of all of the monitor groups for a particular component type such as API groups or agent groups. This screen also displays individual statistics graphs for each of the components in the selected monitor group.
- Monitor Component Details - The third level of screens that display the monitoring details of a single component such as a single API or single agent.

2.3.1 Detecting a Problem

Components that are currently (based on the latest information retrieved during the last refresh) in a problematic state are highlighted in red (alert) through-out the console. The cause of the problematic state depends on the type of component. For example, an API group turns red when a single API inside that group has an average response time over the average response time threshold set for that API.

Additionally, components that are near a problematic state are highlighted in yellow (warning). Again, the cause for the warning state depends on the type of component. For example, an API turns yellow when the average response time is within 80% of the threshold value.

2.3.2 Opening Multiple Monitor Screens at the Same Time

Any combination of the summary tree and graphs or monitor component details screens can be opened at the same time. By opening the appropriate screens, it is possible to focus your attention on specific components if the need arises.

2.4 Restricting Access to the System Management Actions

It is possible to restrict access to certain actions available in the System Management console. Permission can be revoked for the following actions within the System Management console:

- Shutdown, suspend, and resume agent or integration servers
- Clear cache
- Disable cache and change cache limits
- Change threads for an agent or integration server
- Enable and disable trace
- Change monitor thresholds

These permissions are maintained with the user group configuration under the "System" entity in the "Platform" module. For more information about maintaining user groups, see the *Yantra 7x Platform Configuration Guide*.

2.5 Resolving Problems with Application Server Response Time

If an application server's response time is too high, the users of the Application Consoles will have difficulty performing their tasks because of the long delay in screen loads and updates.

If this happens, investigate the following potential causes:

- [Application Server Cache Level](#)
- [Client Load Too High](#)
- [JSPs Not Precompiled](#)
- [Frequent JVM Full Garbage Collection](#)

2.5.1 Application Server Cache Level

An application server may respond slowly if it is using too much cache memory. An application server's cache level can be viewed (and cleared if necessary) on the [Application Server Details](#) screen.

To view the current cache memory used for an application server:

1. Locate the application host running the application server for which you want to view cache information on the [System Management Console](#). Single-click the icon. The [Application Hosts Summary](#) screen displays.

2. Locate the summary graph for the application server. A progress bar appears to the right of the progress bar indicating the amount of cache memory used by that application server. The label next to the progress bar indicates the amount of memory (in kilobytes).

If you then want to clear the cache memory for an application server:

1. Left-click the summary graph. The [Application Server Details](#) screen displays.
2. Click the "Clear Cache" button.

2.5.2 Client Load Too High

If there are many Application Consoles users accessing the application at the same time, there may not be enough application servers running to process all of the requests in a timely manner.

You can get a general idea of the load each application server is handling on the [Application Server Details](#) screen. This screen displays the number of requests being handled by an application over the last 4 hours.

To view the requests handled by an application server:

1. Locate the application host running the application server for which you want to view the requests on the [System Management Console](#). Left-click the icon. The [Application Hosts Summary a](#) screen displays.
2. Locate the summary graph for the application server for which you want to view table level cache information. Left-click the graph. The [Application Server Details](#) screen displays.
3. Select the "# Of Requests" radio button under the graph.

If the application servers are processing too many requests, you can always run more application servers and load balance them to increase the capacity that can be handled. For more information about load balancing, see the *Yantra 7x Performance Management Guide*.

2.5.3 JSPs Not Precompiled

An application server may initially respond slowly if the JSP files used in the Yantra 7x Application Consoles are not precompiled. The first time any user navigates to a screen, the JSP(s) used for the screen are compiled. This compilation process can be time-consuming. Therefore,

Yantra strongly recommends that all JSPs are precompiled during EAR creation. For more information about precompilation, see the *Yantra 7x Performance Management Guide*.

2.5.4 Frequent JVM Full Garbage Collection

JVM full garbage collection is a costly operation that can slow down the application server if it is performed frequently. Garbage collection will be automatically started when the JVM is using most of its allocated memory. Use the `-verbose:gc` parameter on the java command line to see how frequently full garbage collections are occurring.

2.6 Resolving Application Server Down Alerts

When an application server goes down unexpectedly, it should be brought back up immediately to ensure that users of the Application Consoles are not impacted.

If an application server goes down frequently, you should view the application server's log files to investigate possible causes.

2.7 Resolving Problems with API Response Time

To ensure that the through-put of the application is at optimal levels, the APIs must be executing with good response times. API response times also can effect the Yantra 7x Application Consoles users because the Application Consoles fetch and update all information using APIs.

If an API's response time goes above it's threshold, investigate the following possible causes:

- [Slow Response Time for a User Exit](#)
- [Slow Response Time for one JVM](#)

2.7.1 Slow Response Time for a User Exit

If the API has implemented user exits, then it may be the response time of the user exit that is slow. You can view the response time of the user exit invocations for the last 4 hours from the [API Details](#) screen in the System Management console.

To view the response time of an API's implemented user exits:

1. Locate the API group that contains the API on the [System Management Console](#). Left-click the group. The [API Groups Summary](#) screen displays.
2. Locate the summary graph for the API. Left-click the graph. The [API Details](#) screen displays.
3. In the Implemented User Exit panel, select the radio button corresponding to the user exit for which you want to view the response time. The graph displays the minimum, maximum, and average response times of that user exit for the last 4 hours.

If you find that the user exit is responding slowly, investigate how that user exit has been implemented. You may want to trace that user exit to find out more information. For more information on tracing a component, see [Section 2.10, "Tracing a Component"](#).

2.7.2 Slow Response Time for one JVM

An API may be invoked on multiple JVMs. If one JVM has slowed down considerably, the response time of all APIs executed in that JVM increases. Therefore, an API's response time may exceed the threshold due to one JVM responding slowly. The response time by JVM for an API can be viewed on the [API Details](#) screen in the System Management console.

To view the response time by JVM:

1. Locate the API group that contains the API on the [Accessing the System Management Console](#). Left-click the group. The [API Groups Summary](#) screen displays.
2. Locate the summary graph for the API. Left-click the graph. The [API Details](#) screen displays.
3. The "Response Time By JVM" panel displays a list of JVMs and the corresponding response times for that API on each JVM.

If you find that the response time is slow on only one JVM, this suggests that the problem lies in that JVM (not with the API in general). You can view more information about the JVM by selecting it in the list and clicking the "View Details" button.

2.8 Resolving Problems with Agent Server Pending Tasks

To maximize the through-put of the application, the agents must be able to process all of the pending tasks within an acceptable time frame. If an agent is not able to process its tasks fast enough, the pending jobs accumulate and cause a bottleneck in the system.

If the pending jobs threshold for an agent is exceeded, investigate the following potential causes:

- [Agent Server is not Running](#)
- [Agent Server Needs More Threads](#)
- [Errors are Being Generated](#)

2.8.1 Agent Server is not Running

An obvious reason the pending jobs of an agent are increasing is that the agent may not be running at all! Ensure that the agents are scheduled to run at the appropriate times of the day.

2.8.2 Agent Server Needs More Threads

If an agent is not able to process the tasks quickly enough, you can increase the number of threads used by the agent to increase the processing rate. The thread count can be increased in the [Agent and Integration Server Instance Detail](#) screen.

To increase the number of threads used by an agent, follow the steps in [Section 2.21, "Increasing Threads on an Agent or Integration Server Instance"](#).

2.8.3 Errors are Being Generated

An agent may not be performing to its full potential if it is generating errors. The most recent errors generated by an agent can be viewed in the [Agent and Integration Server Detail](#) screen.

To view the most recent errors for an agent:

1. On the [System Management Console](#), locate the agent or integration server group that contains the agent for which you want to view the

most recent errors. Left-click on the group icon. The [Agent and Integration Server Summary](#) screen displays.

2. Locate the summary graph that corresponds to the agent for which you want to view the most recent errors. Left-click on the graph. The [Agent and Integration Server Detail](#) screen displays.
3. The "Most Recent Errors" panel displays the errors for the agent. Take note of the time the errors occurred. If they have occurred recently, investigate the logs to determine the cause of the errors and try to remedy them.

2.9 Resolving Problems with JMS Queue Number of Messages

To ensure that the integration points between Yantra 7x and other systems that are using JMS messaging queues are working properly, the number of messages in the queues should be monitored. If the messages in the queue are not being processed quickly enough or not being processed at all, then something has gone wrong at the integration point. If the problem at this integration point is not resolved quickly, then more problems are sure to happen somewhere further along in the process.

If the threshold for number of messages is exceeded, investigate the following potential causes:

- Message consumer process is not running
- Processing rate is low for message consumer process

2.9.1 Message Consumer Process is not Running

If the messages in the queue are building up, it is possible the process that consumes the messages is not running. Ensure that the message consuming process is scheduled to run at appropriate times.

2.9.2 Processing Rate is Low for Message Consumer Process

Another potential reason for messages in the queue building up, is that the process that consumes the messages is not processing them quickly enough.

If the process that consumes the messages is an agent, you can investigate the possible causes in [Section 2.8, "Resolving Problems with Agent Server Pending Tasks"](#) to resolve the problem.

If the process that consumes the message is one external to the Yantra application, then you should investigate that process.

2.10 Tracing a Component

Tracing is a valuable tool that can be used to investigate various problems while a component is running. When trace is turned on for a component, additional messages are output that will help you to determine the problem.

To trace a component, go to the [Traced Components List](#) available from the main menu of the [System Management Console](#) under Tools > Trace Components.

2.11 Enabling an Agent Trace

To trace an agent, you need to follow the sequence given below:

1. Start your application server.
2. Start the agent server that you wish to trace.
3. To start the trace, go to the [Traced Components List](#) available from the main menu of the [System Management Console](#) under Tools > Trace Components and select Agents.
4. Save and close the trace component list.
5. Trigger the agent.

Note: If the trace component is added before starting the agent server, no trace is found in the log file.

2.12 Stopping a Component Trace

To stop tracing a component, go to the [Traced Components List](#) screen available from the main menu of the [System Management Console](#) under Tools > Trace Components. For more information about stopping trace on a component, see [Section 3.12, "Traced Components List"](#).

2.13 Viewing Application Server Cache Information

There are two levels of cache information available for an application server. The first is the global memory used for cache available on the [Application Hosts Summary a](#) screen. Table level cache information is also available in the [Table Level Cache List](#) screen.

To view global memory used for cache for an application server:

1. Locate the application host running the application server for which you want to view cache information on the [System Management Console](#). Single-click the icon. The [Application Hosts Summary a](#) screen displays.
2. Locate the summary graph for the application server. A progress bar appears to the right of the progress bar indicating the amount of cache memory used by that application server. The label next to the progress bar indicates the amount of memory (in kilobytes).

To view table level cache information for an application server:

1. Locate the application host running the application server for which you want to view table level cache information on the [System Management Console](#). Left-click the icon. The [Application Hosts Summary a](#) screen displays.
2. Locate the summary graph for the application server for which you want to view table level cache information. Left-click the icon. The [Application Server Details](#) screen displays.
3. Click the "Table Level Cache" button. The [Table Level Cache List](#) screen displays.

2.14 Clearing Database Cache for an Application Server

To clear the complete cache memory for an application server:

1. Locate the application host running the application server for which you want to clear the cache on the [System Management Console](#). Left-click the icon. The [Application Hosts Summary a](#) screen displays.

2. Locate the summary graph for the application server for which you want to clear the cache. Left-click the icon. The [Application Server Details](#) screen displays.
3. Click the "Clear Cache" button.

2.15 Disabling Cache or Changing Database Cache Limits for an Application Server

It is possible to disable cache entirely or change the number of cache limit for a specific database table in the [Table Level Cache List](#) screen.

To disable cache or change the cache limit on a database table for an application server:

1. Locate the application host running the application server for which you want to disable or change cache limits on the [System Management Console](#). Left-click the icon. The [Application Hosts Summary a](#) screen displays.
2. Locate the summary graph for the application server for which you want to disable or change cache limits. Left-click the icon. The [Application Server Details](#) screen displays.
3. Click the "Table Level Cache" button. The [Table Level Cache List](#) screen displays.

2.16 Viewing Application Server Properties

To view the properties of an active application server (as specified in the `yfs.properties` file):

1. Locate the application host running the application server for which you want to view the properties of on the [System Management Console](#). Left-click the icon. The [Application Hosts Summary a](#) screen displays.
2. Locate the summary graph for the application server for which you want to view the properties of. Left-click the icon. The [Application Server Details](#) screen displays.
3. Click the "Server Properties" button. The Server Properties screen displays a complete list of all properties and their corresponding values. These properties cannot be modified from within the System

Management console. Any modification to these properties must be made in the appropriate `yfs.properties` file. After making any modifications to this file you must reboot your application server. For more information about these properties see the *Yantra 7x Installation Guide*.

2.17 Shutdown, Suspend, or Resume an Agent or Integration Server Instance

Actively running agents and integration servers can be shutdown, suspended, or resumed from the [Live Agent or Integration Servers](#) screen.

Note: The suspend and resume actions are not supported for the health monitor or any of its sub-services.

To shutdown, suspend, or resume an agent or integration server:

1. From the [System Management Console](#) select Tools > View Live Agents or Integration Servers. The [Live Agent or Integration Servers](#) screen displays.
2. Locate the agent or integration server and click the appropriate button depending on what action you want to perform (shutdown, suspend, or resume). You can select multiple agents, integration servers or a combination of both to perform that action for more than one agent or integration server.

2.18 Starting an Agent Server

The `agentserver` utility starts processes responsible for processing transactions generated by the time-triggered transactions (agents). Start an agent server, using the directions below that apply to your operating system and specify the Agent Server you supplied in the Yantra 7x Configurator for the time-triggered transaction:

- If you are using UNIX, run the `<YFS_HOME>/bin/agentserver.sh` script.

For example:

```
agentserver.sh <agent_server_name>
```

2.19 Starting an Integration Server

To start an instance of a Yantra 7x Integration Server:

- If you are using UNIX, run the `<YFS_HOME>/bin/startIntegrationServer.sh` command

Note: You cannot start the Yantra 5x Integration Server for services using the HTTP or EJBtransport mechanisms.

Note: The following message appears after the WebLogic JMS transaction timeout period has elapsed.

```
<date/time stamp> [Thread-??] ERROR
services.jms.JMSConsumer -Could not
successfully process message
weblogic.jms.common.
TransactionRolledBackException:
```

It can be ignored. There is no harm done and no transactions are rolled back.

Note: When using WebSphere MQ as the messaging system, you may see the following message when the server is started:

```
unable to load message catalog - mqji
```

This message is coming from WebSphere MQ. To prevent this message, the following directory must be present in the classpath for the MQ Client and the Integration Adapter in addition to the WebSphere MQ JAR files:

```
<MQ_JAVA_INSTALL_PATH>/lib.
```

2.20 Scheduling an Agent

The `triggeragent` utility is used for scheduling agents (time-triggered transactions).

You can use any application scheduler provided by your operating system to schedule time-triggered transactions. For example, in a UNIX environment, you can use the CRON command, where `<criteriaID>` is defined for each time-triggered transaction in the Yantra 7x Configurator.

```
triggeragent.sh <criteriaID>
```

You can also run any time-triggered transaction on a one-time basis using this command.

Schedule the `triggeragent.sh` script, located in the `<YFS_HOME>/bin/` directory, to trigger a time-triggered transaction to begin processing.

For more information on the agent criteria for time-triggered transactions see *Yantra 7x Platform Configuration Guide*.

For more information on scheduling time-triggered transaction refer to *Yantra 7x Installation Guide*.

2.21 Increasing Threads on an Agent or Integration Server Instance

Changing the number of threads being used by an actively running agent or integration server instance can be done from the [Agent and Integration Server Instance Detail](#) screen. The changes made on this screen are temporary and are lost if the agent is restarted. To change the threads on an active agent:

1. On the [System Management Console](#), locate the agent or integration server group that contains the agent or integration server for which you want to change the threads. Left-click on the group icon. The [Agent and Integration Server Summary](#) screen displays.
2. Locate the summary graph that corresponds to the agent or integration server for which you want to change the threads. Single-click on the graph. The [Agent and Integration Server Detail](#) screen displays.
3. Locate the server that is running the agent or integration server in the "Servers Running this Agent or Integration Server" panel. Select

the server and choose "View Details". The [Agent and Integration Server Instance Detail](#) screen displays.

To permanently change the threads used when an agent or integration server is started, you must modify the configuration of that agent or integration server. For more information about configuring the initial threads of an agent or integration server see the *Yantra 7x Platform Configuration Guide*.

Note: You cannot change the thread levels for the health monitor.

2.22 Viewing Properties for an Agent or Integration Server Instance

You can view the properties of an actively running agent or integration server from the [Agent and Integration Server Instance Detail](#) screen in the System Management console.

1. On the [System Management Console](#), locate the agent or integration server group that contains the agent or integration server for which you want to view the properties. Left-click on the group icon. The [Agent and Integration Server Summary](#) screen displays.
2. Locate the summary graph that corresponds to the agent or integration server for which you want to view the properties. Single-click on the graph. The [Agent and Integration Server Detail](#) screen displays.
3. Locate the server that is running the agent or integration server in the "Servers Running this Agent or Integration Server" panel. Select the server and choose "View Details". The [Agent and Integration Server Instance Detail](#) screen displays.

2.23 Changing Monitor Groupings

To change the components that have been grouped into a monitor group:

Locate the monitor group you want to change on the [System Management Console](#). Right-click the icon, and select "Modify Monitor Group". The [Monitor Group Details](#) screen displays.

2.24 Setting Notification of an Application Server Shutdown

It is possible to be notified whenever an application server goes down. For information on setting notification for this condition see [Section 2.1.2.2.1, "Alert when Server Goes Down"](#).

2.25 Setting Notification of an Agent or Integration Server Unexpected Termination

It is possible to be notified whenever an agent or integration server terminates unexpectedly. For information on setting notification for this condition see [Section 2.1.4.2.1, "Alert when Agent or Integration Server Terminates Unexpectedly"](#).

2.26 Setting Notification of Threshold Reached or Exceeded

It is possible to be notified whenever a threshold is exceeded. For more information on setting notification for this condition see the following sections:

[Section 2.1.3.1.1, "Alert when the Response Time Threshold is Exceeded"](#)
[Section 2.1.4.3.1, "Alert when the Pending Tasks Threshold is Exceeded"](#)
and [Section 2.1.5.1.1, "Alert when the Number of Messages Threshold is Exceeded"](#).

2.27 Starting the Health Monitor Agent

For information on starting the health monitor agent, see [Section 2.1.1, "Running the Health Monitor Agent"](#).

Part II

System Management Screen Reference

The chapters in this section provide screen and field descriptions for all Inventory Synchronization screens.

This section contains the following chapters:

- [System Management Console](#)
- [Application Hosts Summary a](#)
- [Application Server Details](#)
- [API Groups Summary](#)
- [API Details](#)
- [Agent and Integration Server Summary](#)
- [Agent and Integration Server Detail](#)
- [Agent and Integration Server Instance Detail](#)
- [JMS Queue Summary](#)
- [JMS Queue Details](#)
- [Monitor Group Details](#)
- [Traced Components List](#)
- [Trace Component Details](#)
- [Table Level Cache List](#)
- [Live Agent or Integration Servers](#)

System Management Console Screens

This chapter provides screen and field descriptions for all System Management Console screens.

3.1 System Management Console

The main view of the System Management console displays an overall status of the entire Yantra 7x application.

Each of the three main types of screens refresh with the latest information every 5 minutes. All other screens within the console do NOT refresh with the latest information.

Table 3–1 System Management Console

Actions	
The following actions are accessible from the Tools menu.	
Create Monitoring Group	This action takes you to the Monitor Group Details screen where you can create a new monitor group for an API, Agent, Integration Server, or JMS Queue.
Trace Components	This action takes you to the Traced Components List screen where you can view and add traces.
View Live Agent or Integration Servers	This action takes you to the Live Agent or Integration Servers screen where you can view and control agent or integration servers.

Table 3–1 System Management Console

Actions	
<p>Status Bar</p> <p>The status bar displays in the bottom right corner of the main application window. The status bar contains icons that represent various information about the entire system. Yantra recommends that you keep the status bar visible at all times so that any problems in the application are immediately visible when the icons change.</p>	
Icons	
	The tool tip of this icon displays information about the user that is logged into the System Management Console.
	The tool tip of this icon displays information about the active locale.
	This icon shows the current status of the trace feature within the System Management console. Tracing can be turned on or off by clicking on this icon. When tracing is off, the icon displays a red circle with an "X". When tracing is on, the icon displays a green circle with a check mark. Trace should always be turned off unless you are trying to debug a problem within the System Management console. When trace is turned on, informative messages are output to the Appender configured in the log4jconfig.xml. When the trace is turned on, the screens in the System Management console perform much more slowly than usual.
	This icon represents the state of ALL application hosts. It displays the appropriate color (normal, red, yellow, or disabled) representing the worst state of all application hosts. For example, if all application servers for one host are down, the icon here is disabled. If all application servers are running, but one application server has exceeded its response time threshold, the icon displays in red. If all application servers are running and have no problems, the icon displays normally.

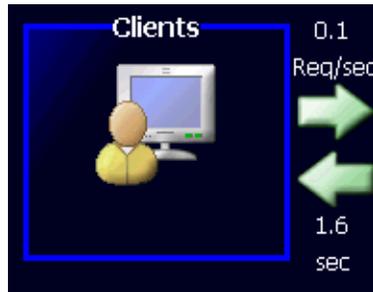
Table 3–1 System Management Console

Actions	
	This icon represents the state of ALL API groups. It displays the appropriate color (normal, red, or yellow) representing the worst state of all API groups.
	This icon represents the state of ALL agent or integration server groups. It displays the appropriate color (normal, red, or yellow) representing the worst state of all agent or integration server groups.
	This icon represents the state of ALL JMS queue groups. It displays the appropriate color (normal, red, or yellow) representing the worst state of all JMS queue groups.

Table 3–2 System Management Console Screen, Clients

Clients

Between the Clients and Application Hosts sections of the screen, the arrows are labeled with information about the requests and responses from the client and servers.



On the arrow going from the client to the application hosts, the number of requests per second is displayed. This number represents the number of HTML requests (per second) coming from users of the Yantra 7x Application Consoles for the last persist interval. This indicates the load on the application servers coming from the users.

On the arrow going from the application hosts to the client, the average response time across all application servers is displayed. This number gives a general idea of the response time that the users of the Yantra 7x Application Consoles are experiencing.

Table 3–3 System Management Console Screen, Application Hosts

<p>Application Hosts</p> <p>For each application host (physical machine) that is running application servers, the following information displays in the Application Hosts panel:</p> <div style="text-align: center;">  </div> <p>The progress bar next to the icon indicates the highest percentage of cache used by any application server running on the host.</p> <p>Next to the progress bar, the highest response time of any application server running on that host is displayed.</p> <p>The name of the application host displays under the icon.</p> <p>To view the Application Hosts Summary screen, single-click (or right-click and select "Details") on the application host.</p>	
<p>Icons</p>	
	<p>Normal - All application servers are running normally with good response times.</p>
	<p>Red - One or more application servers are running with a response time that exceeds the set threshold.</p>
	<p>Yellow - One or more application servers are running with a response time that is close to the set threshold.</p>

Table 3–3 System Management Console Screen, Application Hosts

	<p>Normal with Yellow "X" - Some of the application servers are down. This icon will only appear if there are multiple application servers running on the host, and one or more application server is down while there is at least one application server that is still running.</p>
	<p>Disabled with Red "X" - All application servers are down.</p>

Table 3–4 System Management Console Screen, API Groups

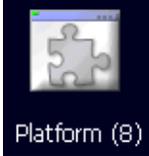
<p>API Groups</p> <p>For each API monitor group, the following information displays in the API Groups panel:</p> <div style="text-align: center;">  </div> <p>The API group name displays under the icon. The number of active APIs in the group is displayed in parenthesis next to the group name. An API is considered to be active if it has been called during the last persist interval.</p> <p>To view the API Groups Summary screen, single-click (or right-click and select "Details") on the API group.</p>	
<p>Icons</p>	
	<p>Normal - All APIs are running normally with good response times.</p>

Table 3–4 System Management Console Screen, API Groups

	<p>Red - One or more APIs are running with a response time that exceeds the set threshold.</p>
	<p>Yellow - One or more APIs are running with a response time that is close to the set threshold.</p>

Table 3–5 System Management Console Screen, Database

<p>Database</p> <p>Yantra 7x System Management does not monitor the database in any way. Yantra recommends using a third-party database monitoring software in addition to the Yantra 7x System Management Console.</p> <p>The Database panel displays information about the database that the application is using.</p>	
Fields	
Driver Name	The name of the database driver used to connect to the database.
Driver Version	The version of the database driver used to connect to the database.
Database Version	The complete database version information is available as a tool tip on the database icon.
Actions	
Clear Cache	This action will clear all cache for the database.

Table 3–6 System Management Console Screen, Agent or Integration Server Groups

<p>Agents or Integration Server Groups</p> <p>For each agent or integration server monitor group, the following information displays in the Agent or Integration Server Groups panel:</p>	
	
<p>The agent or integration server group name displays under the icon. The number of active agents and integration servers in the group is displayed in parenthesis next to the group name. An agent or integration server is considered to be active if it had any activity during the last persist interval.</p> <p>To view the Agent and Integration Server Summary screen, single-click (or right-click and select "Details") on the agent or integration server group.</p>	
Icons	
	<p>Normal - All agents have a number of pending tasks lower than the set threshold.</p>
	<p>Red - One or more agents have a number of pending tasks that exceeds the set threshold.</p>
	<p>Yellow - One or more agents have a number of pending tasks that is close to the set threshold.</p>

Table 3–7 System Management Console Screen, JMS Queue Groups

<p>JMS Queue Groups</p> <p>For each JMS queue monitor group, the following information displays in the JMS Queue Groups panel:</p>	
	
<p>The JMS queue group name displays under the icon.</p> <p>To view the JMS Queue Summary screen, single-click (or right-click and select "Details") on the JMS queue group.</p>	
Icons	
	<p>Normal - All queues have a number of messages lower than the set threshold.</p>
	<p>Red - One or more queues have a number of messages that exceeds the set threshold.</p>
	<p>Yellow - One or more queues have a number of messages that is close to the set threshold.</p>

3.2 Application Hosts Summary a

This screen displays summary-level information about the application hosts.

On the left side of the screen, a tree displays all of the applications hosts with their corresponding application servers under each host. The icon next to each component represents its current state (good, alert, warning, or down). When a particular application host is selected,

summary graphs of all application servers in that host are displayed on the right. Next to each summary graph is a progress bar that represents the amount of cache memory that is being used for that application server.

The summary graphs display the average response time of the application server over the last four hours. If the average response time of the application server is currently over the threshold, then the background of the graph is red.

3.3 Application Server Details

This screen displays detailed information about the application server.

The current status (good, alert, warning, or down) of the application server is represented by the icon next to the Host field.

By default, the graph displays the minimum, maximum, and average response time (in seconds) of the HTML requests in the last 4 hours. The currently set threshold limit for application server response time displays on the graph as a horizontal red line.

To view the number of requests made to the application server over the last 4 hours, select the "# of Requests" radio button under the graph.

Table 3–8 Application Server Details Screen, Primary Info

Fields	
Host	The name of the physical host on which this server is running.
Server ID	The generated ID for the application server.
Server Name	The unique name of the server.
Actions	

Table 3–8 Application Server Details Screen, Primary Info

Server Properties	Select this action to view the application server properties (as specified in the <code>yfs.properties</code> file). The "Server Properties" screen displays a complete list of all properties and their corresponding values. These properties cannot be modified from within the System Management console. Any modification to these properties must be made in the appropriate <code>yfs.properties</code> file. Yantra recommends that all application servers use the same <code>yfs.properties</code> file. For more information about these properties see the <i>Yantra 7x Installation Guide</i> .
Change Threshold	When the "Min, Max, and Average" response time radio button is selected, the application server response time threshold limit displays under the graph on the Application Server Detail screen. To change the threshold, click this button, enter a new threshold value in the dialog that appears, and click "Apply".

Table 3–9 Application Server Details Screen, Cache Info

Fields	
Cache Level	This field displays the current cache memory being used as a progress bar and a value (in kilobytes). The entire cache progress bar represents 100 k of memory. Therefore, if the cache bar is filled 50%, then 50 k of memory is being used. If a high amount of memory is being used for caching within a particular application server, that may affect the application server's performance.
Actions	
Clear Cache	This action will clear all cache for the application server.
Table Level Cache	This action takes you to the Table Level Cache List screen where you can view a finer level of cache information for an application server..

3.4 API Groups Summary

This screen displays a summary level of information about the API groups.

On the left side of the screen, a tree displays all of the API groups with their corresponding APIs under each group. The icon next to each component represents its current state (good, alert, or warning). When a particular API group is selected, summary graphs of all APIs in that group are displayed on the right.

The summary graphs display the average response time of the API over the last four hours. If the average response time of the API is currently over the threshold, the background of the graph is red.

To view detailed information about an API, you can go to the [API Details](#) screen using any of the following methods:

- Single-click on the API graph.
- Right-click on the API graph and select "Details".
- Double-click the API within the tree.
- Right-click on the API within the tree and select "Details".

3.5 API Details

This screen displays detailed information about an API.

The current status (good, alert, or warning) of the API is represented by the icon next to the "API Name" field.

By default, the graph displays the minimum, maximum, and average response time (in seconds) over the last 4 hours. The average response time threshold appears on the graph as a horizontal red line.

To view the number of invocations for that API over the last 4 hours, select the "# Of Invocations" radio button.

Some standard Yantra 7x APIs also record API-specific statistics. For example, the createOrder API records the number of orders and order lines created during each API invocation. If the API that the API Detail screen was opened for contains API-specific statistics, they appear as check boxes under the "API Specific Statistics" radio button. To view the API-specific statistics in the graph, select the "API Specific Statistics"

radio button and then select the appropriate check boxes next to the statistics you would like to see in the graph. It is possible to graph multiple statistics at the same time by checking multiple check boxes.

Table 3–10 API Details Screen, Primary Info

Fields	
API Name	The name of the API.
Trace Status	The current trace status, or "Off" if the API is not currently being traced.
Actions	
Change Trace Status	Select this to change the tracing status of an API. Select a new trace status in the dialog, and click "OK". To turn tracing off, select "Off" in the dialog.
Change Threshold	When the "Min, Max, and Average Response Time" radio button is selected on the API Detail screen, the currently set response time threshold appears under the graph. To change the threshold, click this button, enter a new threshold value in the dialog, and click "Apply".

Table 3–11 API Details Screen, Response Time By JVM

Response Time By JVM	
The JVMs listed here are only the application server JVMs that called this API. This information can be used to determine if the API response time is consistent across all JVMs. If the response time is not consistent, there might be an issue with a specific JVM.	
Fields	
Host	The host name.
Server Name	The server name.
Average	The average response time in seconds.
Minimum	The minimum response time in seconds.
Maximum	The maximum response time in seconds.

Table 3–11 API Details Screen, Response Time By JVM

Actions	
View Details	To open the Application Hosts Summary a screen for a JVM in this list, select the appropriate JVM and click this button.

Table 3–12 API Details Screen, Implemented User Exits

<p>Implemented User Exits</p> <p>Each implemented user exit appears as a radio button above the user exit graph. The statistics for each user exit can be viewed by changing the user exit radio button selections.</p> <p>By default, the user exit graph displays the minimum, maximum, and average response time (in seconds) for the user exit over the last 4 hours. The number of invocations for that user exit over the last 4 hours can also be viewed by selecting the "# Of Invocations" radio button under the user exit graph.</p>	
Actions	
Change Trace Status	Select this to change the tracing status of a user exit. Select a new trace status in the dialog, and click "OK". To turn tracing off, select "Off" in the dialog.

3.6 Agent and Integration Server Summary

This screen displays a summary level of information about the agent or integration server groups.

On the left side of the screen, a tree displays all of the agent or integration server groups with their corresponding agents and integration servers under each group. The icon next to each agent represents its current state (good, alert, or warning). When a particular agent or integration server group is selected, summary graphs of all agents in that group are displayed on top portion of the right side of the screen. Summary graphs for all integration servers in that group are displayed on the bottom portion of the right side of the screen as well.

The summary graphs for the agents display the number of pending tasks for the agent over the last four hours. If the number of pending tasks for the agent is currently over the threshold, the background of the graph is red.

The summary graphs for the integration servers display the processing rate of the integration server over the last four hours.

To view detailed information about an agent, you can go to the [Agent and Integration Server Detail](#) screen using any of the following methods:

- Single-click on the agent graph.
- Right-click on the agent graph and select "Details".
- Double-click the agent within the tree.
- Right-click on the agent within the tree and select "Details".

3.7 Agent and Integration Server Detail

This screen displays detailed information about agents and integration servers. This screen displays differently depending on whether an agent or integration server is being viewed.

The current status (good, alert, or warning) of an agent is represented by the icon next to the "Service Name" field, or next to the "Sub Service Name" field for an integration server.

When viewing an agent, this screen displays the pending tasks for that agent over the last 4 hours by default. The pending tasks threshold appears on the graph as a horizontal red line. To troubleshoot problems with agent pending tasks, see [Section 2.8, "Resolving Problems with Agent Server Pending Tasks"](#).

To view the processing rate (tasks/hour) of the agent over the last 4 hours, select the "Processing Rate" radio button.

Note: Unlike other agents, in the health monitor details the default statistic is "Processing Rate". This implies the graph is plotted for the "Processing Rate" and no data is available for the number of pending tasks.

Some agents also record agent-specific statistics. For example, the scheduleOrder agent records the number of orders backordered and the number of orders scheduled while the agent is running. If the agent that the detail screen was opened for contains agent-specific statistics, they appear as check boxes under the "Agent Specific Stats" radio button. To view the agent-specific statistics in the graph, select the "Agent Specific

Stats" radio button and then select the appropriate check boxes next to the statistics you would like to see in the graph. It is possible to graph multiple statistics at the same time by checking multiple check boxes. For more information on other agent statistics see the time-triggered transaction reference in the *Yantra 7x Platform Configuration Guide*.

When viewing an integration server, this screen displays the processing rate for that integration server over the last 4 hours by default.

To view the minimum, maximum, and average response time (in seconds) for the integration server over the last 4 hours, select the "Min, Max, and Average Response Time" radio button.

Table 3–13 Agent or Integration Server Detail Screen, Agent Primary Info

Agent Server Primary Info Fields	
Service Name	The agent criteria ID for the time triggered transaction.
Server Name	The server on which an instance of the agent is running (or will be run if the agent is not started). For more information about this parameter, see the <i>Yantra 7x Installation Guide</i> .
JMS Queue Name	The name of the JMS queue that contains messages to be processed by this agent.
Initial Context Factory	The class providing an Initial Context implementation for your application server to enable remote Java clients to connect. For more information about initial context factories, see the <i>Yantra 7x Platform Configuration Guide</i> .
QCF Lookup	The name of the queue connection factory. This name corresponds with a JMS connection factory configured in the application server running Yantra 7x.
Provider URL	The URL containing the protocol and address used to access the JMS queue.
Initial # Of Threads	When the agent is initially started, this is the number of concurrent threads this transaction should be run with. The number of threads an agent server is currently using can be changed in the Agent and Integration Server Instance Detail screen.

Table 3–13 Agent or Integration Server Detail Screen, Agent Primary Info

Time Needed For Completion (Min.)	If the agent is processing records (the processing rate is positive and there are pending tasks left), this field displays the estimated time left (in minutes) for the agent to complete all of its pending tasks. This is an estimation that assumes that the processing rate remains constant and the number of pending tasks does not increase.
Pending Jobs Statistics Collected	A flag specifying whether the statistics of the pending jobs were collected. The default value is N. Note: There is no data available for pending jobs in health monitor agents. Hence, the default statistic is processing rate.
Trace Status	The current trace status for this agent is "Off" if the agent is not currently being traced. Note: The trace status is Off for the health monitor with a component name anything other than All. For example, if APIMonitor is the only component in the health monitor trace, then the Trace Status is Off.
Integration Server Primary Info Fields	
Sub Service Name	The unique runtime ID specified when creating this integration service in the service builder. For more information about the service builder see the <i>Yantra 7x Platform Configuration Guide</i> .
Server Name	The server on which an instance of the integration server is running (or will be run if the server is not started). For more information about this parameter, see the <i>Yantra 7x Installation Guide</i> .
Initial # Of Threads	When the integration server is initially started, this is the number of concurrent threads this integration server should be run with. The number of threads an integration server is currently using can be changed.
Trace Status	The current trace for this integration server, or "Off" if the integration server is not currently being traced.
Actions	

Table 3–13 Agent or Integration Server Detail Screen, Agent Primary Info

Change Trace Status	Select this to change the tracing status of an agent or integration server. Select a new trace status in the dialog, and click "OK". To turn tracing off, select "Off" in the dialog.
Change Threshold	When viewing an agent, and the pending tasks radio button is selected, the currently set pending task threshold displays under the graph. To change the threshold, click this button, enter a new threshold value in the dialog, and click "Apply".

Table 3–14 Agent or Integration Server Detail Screen, Servers Running on this Agent or Integration Server

Fields	
Host Name	The name of the physical machine where this server is running.
Server ID	The unique server ID.
# Of Threads	The number of threads used for this agent in the server.
Actions	
View Details	To view the details of the server, select the appropriate server in the list and click this button. This displays the Agent and Integration Server Instance Detail screen.

Table 3–15 Agent or Integration Server Details Screen, Most Recent Errors

Most Recent Errors	
The 10 most recent errors that were generated by the agent or integration server are displayed. These errors may help you resolve and problems that you may be having with the agent or integration server	
Fields	
Code	The error code.

Table 3–15 Agent or Integration Server Details Screen, Most Recent Errors

Date	The date and time the error occurred.
Description	A description of the error.

3.8 Agent and Integration Server Instance Detail

This screen displays information about a server that is currently running.

Table 3–16 Agent or Integration Server Instance Detail Screen, Header

Fields	
Host Name	The name of the physical host where the server process is running.
Server ID	The unique ID for the server.
Server Name	The unique name for the server.
Server Type	The type of the server.
Status	The status of the server. The server can be Active or Suspended.
Actions	
Shutdown Server	Select this to shutdown a server.
Suspend Server	Select this to suspend a server. This action is only available when the server is active. Note: This action is not available for health monitor agents.
Resume Server	Select this to resume a server. This action is only available when the server is suspended. Note: This action is not available for health monitor agents.
Table Level Cache	Select this to view table level cache information of a server. The Table Level Cache List screen appears.

Table 3–16 Agent or Integration Server Instance Detail Screen, Header

Clear Cache	Select this to clear the cache of a server.
View Properties	Select this to view the server properties (as specified in the <code>yfs.properties</code> file). The "Server Properties" screen displays a complete list of all properties and their corresponding values. These properties cannot be modified from within the System Management console. Any modification to these properties must be made in the appropriate <code>yfs.properties</code> file. For more information about these properties, see the <i>Yantra 7x Installation Guide</i> .

Table 3–17 Agent or Integration Server Instance Detail Screen, Agents or Services Being Run By This Server

Fields	
Service Name	The name of the service.
Status	The status of the service. The service can be Active or Suspended.
# Of Threads	The number of threads being used within the service.
Actions	
Suspend	To suspend a service running within a server, select the appropriate service(s) and click "Suspend". This action is only available when the service is active. Note: This action is not available for health monitor agents.
Resume	To resume a service running within a server, select the appropriate service(s) and click "Resume". This action is only available when the service is suspended. Note: This action is not available for health monitor agents.

Table 3–17 Agent or Integration Server Instance Detail Screen, Agents or Services Being Run By This Server

Apply Thread Changes	<p>To change the number of threads used for a service running within a server, double-click the "# Of Threads" cell in the table for the service you want to modify. Enter a new value in the column, and click "Apply Thread Changes".</p> <p>The changes made to the threads on this screen are temporary. When the server is restarted, it uses the initial number of threads configured for the server. For more information on setting the initial thread count, see the <i>Yantra 7x Platform Configuration Guide</i>.</p> <p>Note: This action is not available for health monitor agents.</p>
Close	This action closes the screen.

One server instance can actually run multiple agents or sub-services. For example, two different agent criteria can be configured to run with the same server name. When the server with the name runs, it actually runs both agents.

3.9 JMS Queue Summary

This screen displays a summary level of information about the JMS queue groups.

On the left side of the screen, a tree displays all of the JMS queue groups with their corresponding JMS queues under each group. The icon next to each component represents its current state (good, alert, or warning). When a particular JMS queue group is selected, summary graphs of all JMS queues in that group are displayed on the right.

The summary graphs display the number of messages in the JMS queue over the last four hours. If the number of message in the queue is currently over the threshold, the background of the graph is red.

To view detailed information about an JMS queue, you can go to the [JMS Queue Details](#) screen using any of the following methods:

- Single-click on the JMS queue graph.
- Right-click on the JMS queue graph and select "Details".

- Double-click the JMS queue within the tree.
- Right-click on the JMS queue within the tree and select "Details".

3.10 JMS Queue Details

This screen displays detailed information about the JMS queue being monitored. The fields displayed depend on the queue type.

Table 3–18 JMS Queue Details

Weblogic Fields	
Name	A unique name for the queue. This name is used to uniquely identify this queue in the System Management console. The current status (good, alert, or warning) of the queue being monitored is represented by the icon next to the "Name" field.
Queue Name	The Weblogic name for the queue (not the JNDI name).
Provider URL	The URL containing the protocol and address used to access the JMS queue. Enter the following value:
MQSeries Fields	
Name	A unique name for the queue. This name is used to uniquely identify this queue in the System Management console. The current status (good, alert, or warning) of the queue being monitored is represented by the icon next to the "Name" field.
Queue Name	The MQSeries name for the queue (not the JNDI name).
Queue Manager	The name of the MQ queue manager to which the queue belongs.
Host Name	The host name of the MQSeries server where the specified queue manager resides.
Port	The port number of the MQSeries server available for the specified queue manager.

Table 3–18 JMS Queue Details

Channel Name	The channel name available for clients to connect to the specified queue manager.
Fields	
Queue Depth	The graph displays the statistics collected for the number of messages (queue depth) in that JMS queue over the last 4 hours (if any).
Threshold	The currently set threshold limit also appears on the graph as a horizontal red line.
Actions	
Change Threshold	Under the graph, the current threshold limit is displayed. To change the threshold, click this button, enter a new threshold value in the dialog, and click "Apply".

3.11 Monitor Group Details

This screen is used to create a new monitor group or modify an existing one. A monitor group is a collection of system components that can be monitored in the Yantra 7x System Management Console.

Table 3–19 Monitor Group Details

Fields	
Group Name	The name of the monitor group.
Actions	
Save	This action saves your changes and closes the window.
Cancel	This action closes the window without saving any changes.

Table 3–20 Monitor Group Details Screen, APIs Tab

API Tab

From the API Tab you can add or remove APIs in a monitor group.



The Available panel on the left displays APIs that can be subscribed to the monitor group. The Subscribed panel on the right displays the APIs that are currently subscribed to the monitor group.

Icons

	To add APIs to the group, select the appropriate APIs from the available list and click this button.
	To add APIs to the group, select the appropriate APIs from the subscribed list and click this button.

Fields

API Name	The name of the API.
Is Service	This field indicates whether the API is a custom service. If the API is a standard Yantra 7x API, this column is not checked.

Table 3–20 Monitor Group Details Screen, APIs Tab

Actions	
Show Only Ungrouped APIs	By default, only APIs that are not already part of another monitor group appear in the available list. To see a list of all APIs, unselect this check box under the available list. This allows you to add the same API to multiple monitor groups.

Table 3–21 Monitor Group Details Screen, Agents Tab

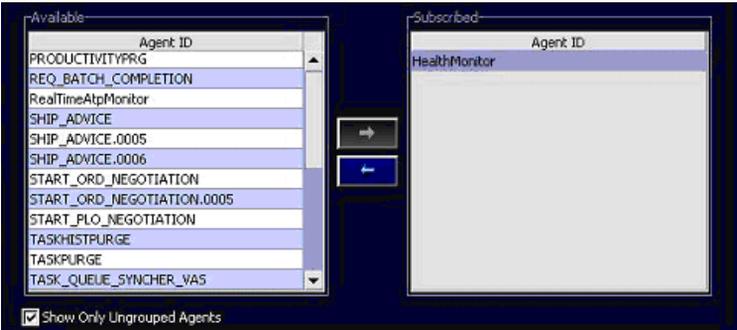
Agents Tab	
From the Agents Tab you can add or remove agents in a monitor group.	
	
The Available panel on the left displays agents that can be subscribed to the monitor group. The Subscribed panel on the right displays the agents that are currently subscribed to the monitor group.	
Icons	
➔	To add agents to the group, select the appropriate agents from the available list and click this button.
➔	To add agents to the group, select the appropriate agents from the subscribed list and click this button.
Fields	
Agent ID	The agent identifier.

Table 3–21 Monitor Group Details Screen, Agents Tab

Actions	
Show Only Ungrouped Agents	By default, only agents that are not already part of another monitor group appear in the available list. To see a list of all agents, unselect this check box under the available list. This allows you to add the same agent to multiple monitor groups.

Table 3–22 Monitor Group Details Screen, JMS Queues Tab

<p>JMS Queues Tab</p> <p>From the JMS Queues Tab you can add or remove JMS queues in a monitor group.</p> <div data-bbox="332 743 1246 1150" data-label="Image"> </div> <p>After making changes to the JMS queues within a monitor group, the health monitor agent must be restarted for the changes to take effect.</p>	
Fields	
JMS Queue Name	The name of the JMS queue.
Actions	
Add	Select this to add a JMS Queue to the monitor group.

Table 3–22 Monitor Group Details Screen, JMS Queues Tab

Modify	Select a JMS queue from the list and select this action to modify values for the JMS queue.
Remove	Select a JMS queue from the list and select this action to remove the JMS queue from the monitor group.

3.12 Traced Components List

Tracing a component is useful when you want more information about a component while the application is running. Typically, tracing is only required when trying to investigate a problem with a component that is either not running properly or is performing with slow response time.

When trace is turned on for a component, additional messages are output to the appropriate places. Where the additional messages are output depends on the logging configuration. For more information about logging, see the *Yantra 7x Installation Guide*.

This screen displays all of the components currently being traced.

Table 3–23 Traced Components List

Fields	
Component Type	The type on component.
Component Name	The name of the component.
Trace Level	The trace level.
Actions	
Add	To trace a component, click this" button. The Trace Component Details screen appears.
Delete	To stop tracing one or more components, select the component(s) in the traced components list and click this button.
Close	Click to close this screen.

3.13 Trace Component Details

This screen is used to start tracing a component.

Table 3–24 Trace Component Details

Fields	
Component Type	<p>Select the type of component you want to trace from one of the following:</p> <p>API - Trace a standard Yantra 7x API.</p> <p>User Exit - Trace an implemented user exit.</p> <p>Service - Trace a service created via the service builder.</p> <p>Agent - Trace an agent.</p> <p>Application Consoles - Trace the user interface layer of the Yantra 7x Application Consoles.</p> <p>Health Monitor - Trace a health monitor.</p>
Component Name	<p>Once a component type has been selected, the available components for that type appear in the Component Name field. Select the component you want to trace from this field.</p> <p>For example, health monitor agent has the following traceable components:</p> <ul style="list-style-type: none"> • JNDICleanup • HealthMonitorPurge • AgentMonitor • APIMonitor • AppServerMonitor • JMSMonitor • HeartbeatMonitor • All

Table 3–24 Trace Component Details

Trace Level	<p>Select the appropriate trace level. The trace level determines what kind of additional information is output. The trace level can be one of the following:</p> <p>Timer - Additional information about the time it took to complete various events. This is useful when trying to resolve performance problems to see what parts of a particular component are taking the longest to complete.</p> <p>SQLDebug - Additional information about the SQL database statements that are executed within the component. This is useful to find out what database statements are actually called. This information can be used to resolve performance problems or to tune your database for optimal performance. For more information about performance tuning, see the <i>Yantra 7x Performance Management Guide</i>.</p> <p>Debug - Some additional miscellaneous information about the component. This information may be useful when trying to debug why a component is not running as expected.</p> <p>Verbose - All available additional information about the component. This trace level outputs the additional information for all other trace levels in addition to any other miscellaneous information that may be available. Select this trace level to get the most information possible for a component.</p> <p>Warning: Tracing a component negatively impacts the performance of that component due to the additional information that needs to be output. It is not recommended to keep a trace on for any components for an extended period of time.</p>
Actions	
Apply	Click to start tracing the component.
Cancel	Click to close the window without saving.

3.14 Table Level Cache List

This screen enables viewing and modifying cache information at a table level for a particular server (application server, agent server, or integration server).

Table 3–25 Table Level Cache List

Field	
Class	The name of the class corresponding to one database table.
Enabled	The check box within this field is checked when caching is enabled for this database table.
Cache Limit	The maximum number of objects that are cached for this database table.
Logical Reads	The number of reads that were successfully retrieved from the cache.
Physical Reads	The number of reads that were not retrieved from the cache.
Hit Ratio	The hit ratio indicates how well the caching is working for this database table. Ideally, the cache ratio should be close to 1. If the cache ratio is close to 0, then the objects being cache are not being accessed frequently. This number is calculated by taking the total number of logical reads and dividing it by the total number of all reads (logical and physical).
# Of Selects	The number of select clauses being cached.
# Of Lists	The number of list clauses being cached.
Cache Clear Count	The number of times cache was cleared for this table. Frequent cache clears could reflect some problem with the access being made to this table.
Objects Cached	The total number of objects currently in the cache.
Action	
Enable Cache	To enable caching on a specific database table, select the appropriate database class and click this button. This action is only enabled when caching is disabled for the selected table.
Disable Cache	To disable caching on a specific database table, select the appropriate database class and click this button. This action is only enabled when caching is enabled for the selected table.

Table 3–25 Table Level Cache List

Clear Cache	To clear the cache memory for a specific database table, select the appropriate database class and click this button.
Modify Cache Limit	To modify the cache limit for a specific database table, select the appropriate database class and click this button. Enter the new cache limit in the dialog and click "OK".
Close	Click to close the window.

Note: Any modifications made in this screen are temporary. If the applicable server is restarted, the changes made in this screen are lost. If you need to permanently enable, disable, or change the cache limit for a specific database table, then you must modify the yfs.properties settings. For more information about yfs.properties, see the *Yantra 7x Installation Guide*.

3.15 Live Agent or Integration Servers

This screen displays all agent and integration servers that are currently running.

Table 3–26 Live Agent or Integration Servers

Fields	
Server ID	The unique ID for the server.
Server Name	The unique name for the server.
Status	The status of the server. The status can be active, suspended, or shutdown in progress.
Actions	

Table 3–26 Live Agent or Integration Servers

Shutdown Server	To shutdown an agent or integration server on the Live Agent or Integration Servers screen, select the appropriate server in the list and click this button. It is possible to select multiple servers in the list to shutdown more than one at a time.
Suspend Server	To suspend a server, select the appropriate server in the list and click this button. This action is only available for servers that are active. Note: This action is not available for health monitor agents.
Resume Server	To resume a server, select the appropriate server in the list and click this button. This action is only available for servers that are suspended. Note: This action is not available for health monitor agents.
Close	Click to close the window.

A

- agents, 3, 25
 - alerts, 14
 - definition, 3
 - details, 53
 - errors generated, 25
 - instance details, 57
 - live, 69
 - monitoring, 13
 - most recent errors, 25
 - other statistics, 15
 - pending tasks, 15
 - alerts, 15
 - changing thresholds, 15
 - resolving problems, 25
 - threshold default value, 15
 - resuming, 30
 - running, 30
 - scheduling, 32
 - server heartbeats, 14
 - shutting down, 30
 - summary, 52
 - suspending, 30
 - threads, 25
 - time-triggered transactions, 3
- alert
 - when number of messages exceeded, 34
 - when pending tasks exceeded, 34
- alerts
 - data published, 17
 - termination attributes, 18
 - threshold exceeded alert attributes, 17
 - when agent or integration server terminates, 34
 - when API response time exceeded, 34
 - when application server down, 34
- API Groups
 - summary, 50
- APIs (Application Programming Interfaces), 2
 - custom APIs, 3
 - definition, 2
 - details, 50
 - invocations, 12
 - monitor groups, 44
 - monitoring, 11
 - other statistics, 12
 - persist intervals, 12
 - response time
 - alerts, 11
 - response times, 11
 - changing thresholds, 12
 - default value, 12
 - resolving problems, 23
 - user exits, 3
- application hosts, 43
 - monitoring, 9
 - raising alerts, 10
 - server heartbeats, 10
 - server names, 10
 - summary, 47
- application logic, 2
- application programming interfaces. See APIs
- application servers
 - cache levels, 21
 - changing cache limits, 29
 - clearing cache
 - cache clearing, 28

- definition, 2
- details, 48
- disabling cache, 29
- raising alerts
 - exceeded threshold, 10
- requests, 2
- resolving server down alerts, 23
- response times, 10
 - changing, 11
 - resolving problems, 21
 - threshold default value, 11
- viewing cache, 28
- viewing properties, 29

Apply Thread Changes action, 59

B

BEA Weblogic, 2, 4

C

cache

- application servers, 21, 28
- cache limit, 29
- Cache Limit field, 68
- changing limits, 29
- clearing, 28, 68
- disabling, 29
- table level cache list, 67
- viewing information, 28

Cache Clear Count field, 68

Cache Level field, 49

Change Threshold action, 49

Change Trace Status action, 51

Channel Name field, 61

Clear Cache action, 22, 49, 69

clients, 2, 22, 42

- definition, 2
- loads, 22

Component Name field, 65

Component Type field, 65, 66

components

- stopping traces, 27
- trace details, 65
- trace list, 65

- tracing, 27

Create Monitoring Group action, 39

D

databases, 4, 45

Disable Cache action, 68

E

Enable Cache action, 68

H

Health Monitor Agent

- alerts
 - data published, 17
 - running, 9, 34

Hit Ratio field, 68

Host Name field, 60

hosts

- definition, 2

I

IBM Websphere, 4

Implemented User Exits fields, 52

Initial Context Factory field, 54

Initial # Of Threads field, 54

integration servers

- alerts, 14
- asynchronous services, 4
- definition, 3
- definition, 4
- details, 53
- instance details, 57
- live, 69
- monitoring, 13
- resuming, 30
- running, 31
- server heartbeats, 14
- shutting down, 30
- summary, 52
- suspending, 30

integration service definitions, 4
intergration servers
 statistics, 16
Inventory Console, 39
Is Service field, 62

J

java virtual memory. See JVM
JMS Queue Name field, 54
JMS queues, 26
JMS (Java Messaging Service) queues, 4
 defintion, 4
 details, 60
 number of messages, 16
 summary, 59
JMS (Java Messaing Service) queues
 monitoring, 16
 number of messages
 alerts, 16
 resolving problems, 26
JSPs
 precompiling, 22
JVMs (Java Virtual Memory)
 full garbage collection, 23
 response times
 slow response times, 24

L

Logical Reads field, 68

M

message consumer processes, 26
Modify Cache Limit action, 69
monitor groups, 18
 agents, 46
 details, 61
 integration servers, 46
 JMS queues, 47
 modifying, 33
monitor screens
 opening multiple screens, 20

N

number of messages
 resolving problems, 26

P

pending tasks
 resolving problems, 25
Physical Reads field, 68
Port field, 60
processing rates, 26
Provider URL field, 54

Q

QCF Lookup field, 54
Queue Depth field, 61
Queue Manager field, 60

R

response time
 resolving problems, 21, 23
Response Time By JVM fields, 51
response times
 APIs, 23
 application server, 21
Resume action, 58
Resume Server action, 57

S

server heartbeats, 14
Server Name field, 55
server properties, 29, 33
Server Properties action, 29
Service Definition Framework, 3, 4
Service Defintion Framework
 JMS Queues, 4
services, 3
Show Only Ungrouped Agents action, 64
Show Only Ungrouped APIs action, 63
Shutdown Server action, 57

- Sub Service Name field, 55
- Suspend action, 58
- Suspend Server action, 57
- System Management Console
 - accessing, 19
 - actions
 - restricting access, 20
 - components, 19
 - detecting problems, 20
 - navigating, 19

T

- Table Level Cache action, 49, 57
- threads
 - changing, 32
- Time Needed For Completion (Min.) field, 55
- time-triggered transactions
 - business process transactions, 3
 - definition, 3
 - monitors, 3
 - purges, 3
- Trace Components action, 39
- trace levels
 - debug, 67
 - SQLdebug, 67
 - timer, 67
 - verbose, 67
- Trace Status field, 51, 55
- Tracel Level field, 67
- tracing
 - stop tracing a component, 27
 - tracing a component, 27

U

- user exits
 - definition, 3
 - response times
 - slow response times, 23
 - statistics, 13

V

- View Live Agent or Integration Servers action, 39

Y

- Yantra 7x
 - monitoring, 9

Z

- # Of Selects field, 68