

WebXMTM

**PLANNING AND
INSTALLATION GUIDE**

Watchfire® WebXM™ 4.5
Planning and Installation Guide (Revision 1, SP5)

Copyright ©1996-2006 Watchfire Corporation. All rights reserved.

No part of this manual may be reproduced in any form or by any means, without permission in writing from Watchfire Corporation.

Watchfire Corporation assumes no responsibility or liability for any errors or inaccuracies that may appear in this manual.

Watchfire, WebCPO, WebXM, WebQA, Watchfire Enterprise Solution, WebXACT, Linkbot, Macrobot, Metabot, Bobby, Sanctum, AppScan, the Sanctum Logo, the Bobby Logo and the Flame Logo are trademarks or registered trademarks of Watchfire Corporation.

Windows, Windows NT, Internet Explorer, Microsoft Data Access Components, Microsoft Internet Information Server are registered trademarks or trademarks of Microsoft Corporation.

All other company or product names are trademarks of their respective owners.

Contents

Introduction

	Chapter 1
WebXM: An Overview	1-2
A Note About Terminology	1-3
What's in This Guide	1-3
WebXM Documentation Suite	1-4
Components of WebXM	1-5
WebXM Control Center	1-5
WebXM Database	1-6
Agent Service and Agents	1-6
Alerting Service	1-7
WebXM Desktop	1-7
Contacting Watchfire	1-8
Technical Support	1-8
Contact Methods	1-9
Email	1-9
Telephone and Fax	1-9
Website	1-9

Planning for WebXM

	Chapter 2
Overview	2-2
How Do You Plan to Use WebXM?	2-2
WebXM Server	2-2
WebXM Desktop	2-4
What You Will Need: WebXM Requirements	2-4
Required WebXM Server Components	2-4
System Requirements — WebXM Server	2-5
Hardware Requirements	2-5
Additional Hardware Requirements	2-6
Operating System and Browser Requirements	2-6
Additional Software Requirements	2-8
System Requirements — WebXM Database	2-8
System Requirements — WebXM Desktop	2-9
Hardware Requirements	2-9
Operating System and Other Software	2-10
User Account Information	2-10
Local System User Account	2-10
Service Account	2-10
What Is Your Existing Environment?	2-12

Website Structure	2-12
Organizational Structure	2-12
Physical Environment	2-13
Single Server Versus Multiple Servers	2-13
WebXM on a Single Server	2-14
WebXM on Multiple Servers	2-14
Multiple Instances of WebXM on a Single Server	2-16
Multiple Instances of the WebXM Control Center	2-17
Multiple Instances of WebXM Agents	2-18

Installing WebXM

Chapter 3

Install Summary	3-2
Preinstall Checklist	3-2
Preparing for Installation	3-6
Information to Have on Hand	3-6
System Requirements	3-7
Preparing for WebXM Server Installation	3-7
Installing Language Packs for International Website Scanning	3-7
Preparing for WebXM Desktop Installation	3-8
Installing WebXM Server	3-8
Installing WebXM on a Single Server	3-9
Installing WebXM on Multiple Servers	3-10
Installing Additional Agents	3-12
Installing Multiple Instances of WebXM on a Single Server	3-13
Configuring the WebXM Server Components	3-14
Modifying the Configuration of WebXM Server	3-17
Activating the Optional Modules	3-17
Configuring NetGenesis Database for Traffic	3-18
Installing MSDE	3-19
Installing WebXM Desktop	3-20
Upgrading WebXM — A Checklist	3-21
SiteMinder and WebXM Integration	3-22
Pre-authorized Authentication	3-22
WebXM Desktop Support	3-23

Implementing the Analytics Module

Chapter 4

Introduction	4-2
Architecture of an Analytics Deployment	4-2
Scaling an Analytics Deployment	4-5
Multiple Instances of WebXM	4-6
Guidelines for Load Balancing	4-6
Data Tag Syntax and Examples	4-6
Form Syntax for all Enabled Pages	4-7
Data Tag Parameters for wf_identityForm	4-8
Form Syntax for Marketing Campaigns	4-10
Restrictions for all Campaign Methods	4-12

A Campaign With a Referring URL	4-12
A Campaign With a Visit to a Landing Page	4-12
A Campaign Having a Query String	4-13
Form Syntax for the Search Results Page	4-14
Data Tag Parameters for Search Results	4-15
JavaScript Syntax for all Pages	4-15
Special Characters in a Data Tag	4-16
Example: Encoded Character in wf_identityForm	4-16
Methods for Populating a Website With Data Tags	4-17
Using Automatic Data tags	4-17
Changing How Content is Automatically Generated	4-17
Examples for Automatic Data Tags	4-19
Job Properties Affected by Data Tags	4-19
If You Have a Defined Path Structure	4-20
If You Have no Path Structure	4-20
Deploying Analytics on Your Website: a Summary	4-20

The WebXM Database

Chapter 5

Overview	5-2
Estimating the Ultimate Size of the WebXM Database	5-2
Database Security	5-3
Using a SQL Server Database with WebXM	5-4
SQL Server Configuration	5-5
Memory	5-5
Security	5-6
Processor	5-6
Server Settings	5-6
Database Properties	5-7
Full-text Search Indexing	5-7
Database and SQL Log File Configuration	5-8
SQL Server Database Backup and Maintenance	5-8
Backup Strategy	5-8
Database Recovery	5-9
Database Maintenance	5-9
Shrinking the Database	5-9
Disk Defragmentation	5-9
Monitoring Database Performance	5-9
Windows Performance Monitor (Perfmon)	5-9
Processes to Monitor	5-10
Troubleshooting SQL Server Database Issues	5-10
SQL Server, Windows and WebXM Tools	5-10
Common Problems and their Symptoms	5-10
Process Blocking	5-11
Using an Oracle Database with WebXM	5-11
Oracle Transaction Type	5-12
Oracle Database Server Installation	5-13
Client Requirements	5-14

Oracle Database Configuration	5-15
Controlling Dynamic Tables in Oracle Databases	5-15
Characterization of Database Growth	5-16
Oracle Database Backup and Maintenance	5-16
Backup Strategy	5-16
Database Recovery	5-16
Database Maintenance	5-16
Disk Defragmentation	5-17
Moving an Oracle Database with WebXM	5-17
Monitoring Database Performance	5-17
Windows Performance Monitor (Perfmon)	5-17
Processes to Monitor	5-17
Troubleshooting Oracle Database Issues	5-18
Oracle, Windows, and WebXM Tools	5-18

Tips, Tricks, and Troubleshooting

Chapter 6

Overview	6-2
Tips & Tricks	6-2
Upgrading WebXM	6-2
Custom Options.xml	6-2
WebXM Server and WebXM Desktop on a Single Machine	6-2
Performance Tuning	6-2
WebXM Server and WebXM Desktop	6-2
WebXM Database	6-4
Verifying the Agent Service and Alerting Service Installation	6-4
Configuring Internet Explorer	6-5
Configuring Internet Explorer for Advanced Login	6-5
Recommended Settings for IIS 6.0 “Web Garden”	6-7
Win2003 and SQL	6-7
Moving an Oracle Database	6-7
Q & A	6-8
Troubleshooting WebXM	6-10

Index

INTRODUCTION

Topics

- WebXM: An Overview
- What's in This Guide
- WebXM Documentation Suite
- Components of WebXM
- Contacting Watchfire

WebXM: An Overview

WebXM is an enterprise-level website testing, monitoring, and reporting software suite. It scans a website, storing the information in a database, and then displays the results in browser-based reports.

There are a number of ways in which you can install WebXM, depending on a variety of factors. This guide helps you determine the best way to install WebXM, based on your requirements, and then describes the installation procedures.

On the server side, WebXM consists of a core engine, which handles reporting, scanning, scheduling, database storage, administration, and visualization of results.

Accessibility, an optional server side module, allows you to test your web pages for compliance with website accessibility standards.

Analytics, an optional server side module, helps you understand how visitors interact with website content so you can measure the effectiveness of that content over a period of time. Its reports provide information about your visitor, their overall experience with the site, and the technologies they embrace. Its web analytics data can be integrated with reports from other modules (as columnar data) and rolled into a My Watchfire dashboard.

Compliance, an optional server side module, reports on regulatory infractions due to security glitches.

Inventory, an optional server side module, gathers information about your website infrastructure which provides you with a technology map of your entire web presence, including important information about each technology component.

Privacy, an optional server side module, is a comprehensive privacy auditing tool that allows you to examine your website for Internet privacy compliance issues.

Quality, an optional server module, allows you to identify issues that affect your overall website quality.

Security, an optional server module, reports about known security vulnerabilities. The module uses detailed security algorithms from Watchfire's AppScan technology as it scans and tests your website.

Traffic, an optional server side plug-in, is a traffic analysis tool that allows you to analyze the behavior of visitors to your website.

WebXM Desktop is a set of Windows applications — Content and Interaction — that handle content scanning and script recording and playback. They can operate independently or exchange jobs with the WebXM Control Center.

A Note About Terminology

For convenience, this guide refers to the set of WebXM components that are installed on the server as WebXM Server. The group of applications, including WebXM Server and WebXM Desktop, is referred to as WebXM.

What's in This Guide

This guide contains the following information:

- **Introduction:** provides general information about WebXM
- **Planning for WebXM:** outlines the factors to take into consideration when determining the configuration that meets your needs
- **Installing WebXM:** describes the steps required to install, upgrade, and configure WebXM
- **Implementing the Analytics Module:** describes the architecture of an analytics deployment, the technologies involved in it, and provides a summary of the steps involved
- **The WebXM Database:** provides information about the database and outlines best practices for configuration and maintenance
- **Tips, Tricks, and Troubleshooting:** provides information to help you solve problems and optimize scan performance

WebXM Documentation Suite

The WebXM documentation suite comprises the following:

Documentation	Users	Description
WebXM Desktop User Guide	WebXM Desktop users	Procedures and concepts relating to WebXM Desktop, Content, and Interaction.
WebXM Desktop Online Help	WebXM Desktop users	Procedures, concepts and context-sensitive help relating to WebXM Desktop
WebXM User Guide	All WebXM users	Procedures and concepts relating to the WebXM Control Center
WebXM Online Help	All WebXM users	Procedures, concepts and context-sensitive help relating to WebXM Control Center
WebXM Administration Guide	System Administrator	Procedures and concepts relating to software administration and configuration issues
WebXM Administration Online Help	System Administrator	Procedures, concepts and context-sensitive help relating to software administration and configuration issues.
Planning and Installation Guide	System Administrator	Describes how to plan for and install WebXM in your organization.

Components of WebXM

This section describes the components that comprise WebXM.

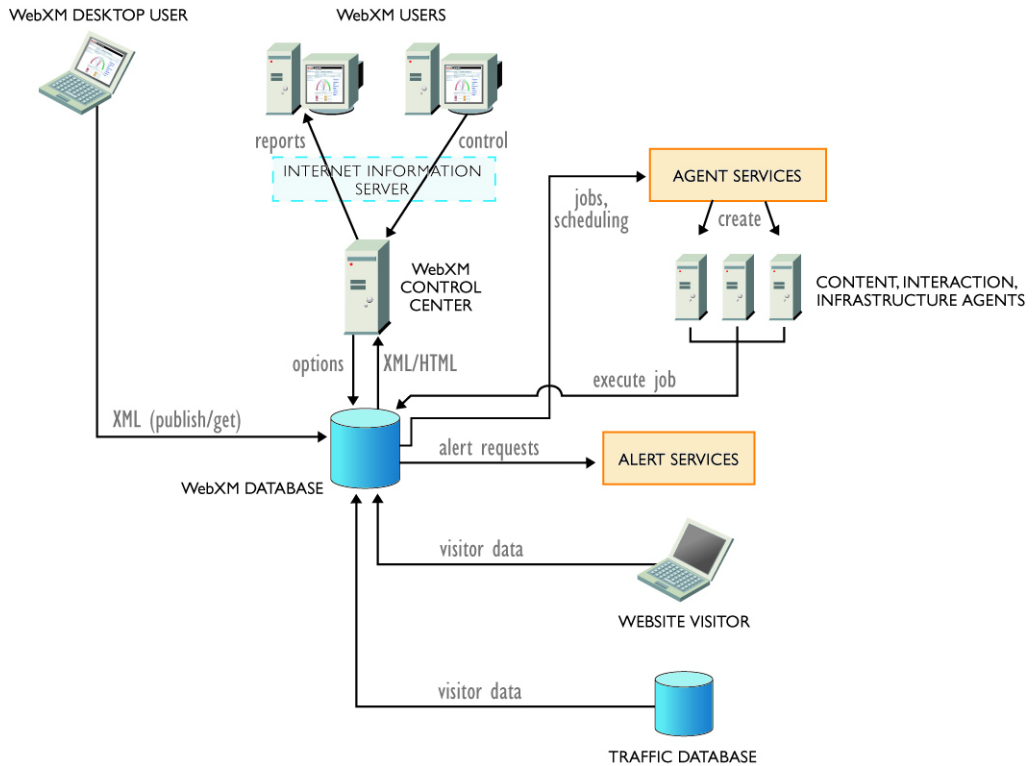


Figure 1-1 WebXM Components

WebXM Control Center

The WebXM Control Center is a set of Active Server Pages that provide the user interface and reports via a web browser. It is the main user interface for WebXM Server and supports administration, development, and reporting.

Depending on your requirements, you can install one WebXM Control Center on a server, or install more than one instance of the WebXM Control Center on a single server. For more information, see “Installing Multiple Instances of WebXM on a Single Server” on page 3-13.

WebXM Database

The WebXM database is the central repository for all information gathered during a job. Each WebXM Control Center that you install has its own corresponding database.

You can use a Microsoft SQL Server 2000 database, an Oracle 9i or Oracle 10g database, or an MSDE database; the one you choose depends on how you plan to implement WebXM.

The WebXM database contains the following data:

- All data gathered by the agents
- Information about the regional or metatag scope applied to report data
- Summarized historical reporting data
- Agent configuration, scheduling, status, and alerting information
- User configuration and permission information
- User customization information

When you install WebXM Server and run the Configuration Wizard, you can choose to connect to an existing database or create a new one. For more information, see “Configuring the WebXM Server Components” on page 3-14.

For information about configuring and maintaining the database, see “The WebXM Database” on page 5-1.

Agent Service and Agents

The agent service monitors the WebXM database for jobs to perform. You can install and run agent services on a single server or on multiple servers.

An agent is a Windows process that is spawned by an agent service when there is a job to be performed. While the job is in progress, the agent records the scan information in the WebXM database. If alerts have been configured, the agent informs the relevant users when specific events occur during the job.

WebXM has the following types of agents: analytics agents (of which there are three), content agents, infrastructure agents, and interaction agents. Together, the analytics agents collect and analyze visitor data

obtained from your websites. Content agents scan your websites and FTP sites for a wide range of issues. Infrastructure agents are used to help create an inventory of your website's technology infrastructure. Interaction agents run scripts that simulate a visitor's session.

Content, interaction, and infrastructure agents can perform only one job at a time; however, a single server can run more than one agent simultaneously. This means that more than one job of the same type may be executed simultaneously on a given machine, with each job being run in its own agent process. (The default is four content agents, four interaction agents, and one infrastructure agent.)

You can also install more than one instance of an agent on a single server, each accessing a different database. This allows you to run a number of jobs, accessing different databases. For more information, see "Installing Multiple Instances of WebXM on a Single Server" on page 3-13.

Alerting Service

The alerting service is responsible for sending alerts, created by content and interaction agents, to the appropriate notification devices. While you can have as many agents and agent services as you need, only one alerting service can be installed for each WebXM database.

WebXM Desktop

The WebXM Desktop is a set of Windows applications — Content and Interaction — that can operate independently or can exchange information with the WebXM Control Center.

WebXM Desktop can be installed from the WebXM CD or downloaded from the WebXM Control Center. For more information, see "Installing WebXM Desktop" on page 3-20.

The WebXM Desktop applications can be used by:

- Individual developers to correct problems identified by jobs on the web interface
- Individual developers to check work to be submitted to a staging server
- Website managers to create or modify content and interaction jobs to be run from the web interface

After you have found areas of your website that need further investigation or repair, you can use the WebXM Desktop applications to do further analysis. Together, they will help to identify the problems and assist you in making the repairs.

You can use the WebXM Desktop applications to support operations in the WebXM Control Center and carry out additional detailed operations to identify problem areas.

Content

Content uses the same scan engine as the WebXM Control Center. You can use it to modify the properties of scan jobs already defined in the Control Center or create new scan jobs that you can run from the Windows workstation or from the Control Center. You can also perform scans on a local file system.

Interaction

You use Interaction to record scripts that duplicate the actions that a user can take on a website. This allows you to ensure that the website is working as designed.

The scripts that you record in Interaction can be run in the WebXM Control Center. You must use Interaction to create the scripts; you cannot create them in the WebXM Control Center.

You publish the scripts that you create to an existing webspace on the WebXM Control Center and then run them from there. You can also get existing scripts from the WebXM Control Center to the WebXM Desktop and then modify them.

Contacting Watchfire

Refer to Support Process for the new IBM® Rational® AppScan® and IBM® Rational® Policy Tester® (formerly Watchfire WebXM) Products. The support channels are different depending on how you obtained the product. Please read the document carefully.

PLANNING FOR WEBXM

Topics

- Overview
- How Do You Plan to Use WebXM?
- What You Will Need: WebXM Requirements
- What Is Your Existing Environment?
- Single Server Versus Multiple Servers

Overview

This section outlines the planning phase of a WebXM installation and the corresponding tasks to be performed. This information will help you determine the best configuration for the WebXM installation.

The configuration you use to set up WebXM depends on a number of factors: what you plan to do with the software, how your organization and website are structured, and how the information is to be distributed.

There are three main steps in the planning phase:

Step 1 - Decide how you plan to use WebXM

Step 2 - Determine what you need

Step 3 - Assess your existing environment

How Do You Plan to Use WebXM?

Before you can begin the WebXM installation, you need to understand how you want to use the software. What will you do with WebXM Server? Do you also need WebXM Desktop and, if so, how will you use it?

The answers to these questions, and others posed in the following sections, will help you identify the components of WebXM you need to install and how you might install them.

WebXM Server

What do you plan to do with the software? Will you be scanning content or running interaction test jobs? What will you be scanning? Your whole website, or a portion of it? How frequently will you be running jobs?

How many people will be running jobs and generating reports? How many people in your organization will need to view reports?

If you plan to scan content, how large is the scan area, and what kinds of information do you want to collect? Do you intend to parse Flash files or execute JavaScript encountered during a scan? What types of reports are required by you and others in your organization?

If you plan to run scripts that test your website, will you be scheduling them to run at regular intervals? Will you be using script iterations? What reports are required?

The types of jobs you expect to run, the size and frequency of those jobs, and the kind of reports you want to generate, are all factors that affect how you should install WebXM. For example, you may decide you need to install additional content agents and interaction agents on multiple servers in order to handle the anticipated volume of scans.

Is your organization structured in units that operate independently of each other? In this case, you may want to consider installing multiple instances of WebXM, one for each unit, with its own dedicated database. Or, do you need a number of agents, on a single server, each accessing a different database?

Will you be testing the pages on your website for compliance with the W3C Web Content Accessibility Guidelines (WCAG) or U.S. Section 508 Standard guidelines? Collecting accessibility data can have an impact on the time taken to complete a scan and the amount of space required to store the results, depending on the particular guidelines you are checking, and whether you plan to collect fragments of the HTML code that contains the issues. While the code fragments can be helpful pieces of information, enabling the collection of code fragments can affect scalability since each page scanned will use more space in the WebXM database.

What about privacy? Does your organization have privacy standards that must be maintained?

What about traffic analysis? Do you currently collect traffic data? Do you plan to? If so, how frequently to you expect to integrate this data with your content scans? Will you integrate with every scan, or only some of them?

Do you have the security module licensed? If so, the database will require more disk space for session file data and test results. The content agent server will also require more disk space in order to perform security tests.

All these issues can impact the type of database you choose to use with WebXM. For example, if you need to be able to store large amounts of scan data, integrated with traffic data, and generate and distribute many different reports, you will need either a SQL Server database or an Oracle database. However, if you are installing WebXM Server in a smaller workgroup, on a single machine, an MSDE database may be sufficient. For more information, see “Single Server Versus Multiple Servers” on page 2-13.

WebXM Desktop

How are the pages on your website currently designed and tested? Do you need to provide developers with the tools to test their work before it is posted to the live server?

Will you be testing the pages on your website for compliance with the W3C Web Content Accessibility Guidelines (WCAG) or U.S. Section 508 Standard guidelines? As with WebXM server, collecting accessibility data can have an impact on the time taken to complete a scan and the amount of space required to store the results, depending on the particular guidelines you are checking, and whether you plan to collect fragments of the HTML code that contains the issues. Enabling the collection of code fragments will adversely affect scalability since each page scanned will use more space in the RAM database.

Would you like to be able to define scan options, test them in a contained environment to ensure that you get required results, and then apply those settings to a larger scan?

The structure of your website and organization will determine how many developers install WebXM Desktop. For example, you may identify developers working on specific areas of your website who should be using WebXM Desktop.

What You Will Need: WebXM Requirements

This section provides information about the components you will need to install, based on the decisions you made about how you want to use WebXM.

Required WebXM Server Components

In order to perform scans and generate reports, you must install the following components:

- WebXM Control Center
- WebXM Content Agent
- WebXM Infrastructure agent
- WebXM Interaction Agent
- WebXM Alerting Service

You will also need a database; it can be SQL Server 2000, Oracle 9i, Oracle 10g, or MSDE 2000, depending on how you choose to install WebXM. For more information, see “The WebXM Database” on page 5-1.

You can install these components on a single server, or across a number of different servers. For sample configurations, see “Single Server Versus Multiple Servers” on page 2-13.

System Requirements – WebXM Server

The following tables provide a summary of the minimum hardware and software required to run WebXM Server.

Hardware Requirements

The systems on which you plan to install the various WebXM Server components should meet the minimum requirements described in Table 2-1.

The minimum hardware requirements are sufficient for installing and running WebXM; however, Watchfire recommends that you expand the hardware requirements to optimize performance. The requirements can be expanded based on the size of your website and the number of WebXM users you will have. Watchfire’s Professional Services will be pleased to assist you in determining what hardware requirements would best suit your needs.

Hardware	Minimum Requirement
Processor	Pentium Class PC (P4 - 1.8 GHz or greater)
Memory	512 MB RAM
Disk Space	200 MB (10 GB of free space on the system hosting the database, 80 GB if security analysis is being performed)

Table 2-1 Hardware Requirements for WebXM Server

Additional Hardware Requirements

Depending on the size of your website or the number of users you have, you may want to consider the following requirements:

- Separate systems for content agent, database server, and Control Center Server
- Separate systems for the Analytics collection agent, processing agent, database server, and Control Center Server
- Configure the database server as a server class machine with a Pentium Class PIII - 1 GHz dual processor and a SCSI disk array configured as RAID(0)

If you are running content scans with security enabled you should consider using a P4 class processor (1.8 GHz) with 5 GB of free disk space and 512 MB of RAM.

Operating System and Browser Requirements

The operating system and browser requirements vary, depending on the individual component. Tables 2-2 to 2-4 summarize the minimum requirements for each of the WebXM Server components, which are themselves described in “Components of WebXM” on page 1-5.



If the website being scanned uses technologies such as Flash, Windows Media, and additional character sets, these technologies must also be installed on the agent server machines.

WebXM Control Center

Table 2-2 lists the minimum operating systems and browsers required for the WebXM Control Center that you plan to use for administration, job and report creation, and viewing.

Software	Version
Operating System	Windows 2000 Professional SP4, Windows 2000 Server SP4, and Windows 2000 Advanced Server SP4 Windows 2003 Server (also install ASP and ASP.NET) Windows XP SP2
Web Server	IIS 5.0, 5.1, or 6.0
Browser	Internet Explorer 6.0 SP1

Table 2-2 WebXM Control Center Requirements



If you install WebXM on Windows 2000 Professional, you may get connection issues related to licensing since this operating system allows a limited number of connections. This configuration is more suited for small workgroups and can be used with an MSDE database.

Table 2-3 lists the minimum operating systems and browsers required for the workstations that you plan to use for report viewing.

Software	Version
Operating System	Windows 2000 Professional SP4
	Windows 2000 Server SP4
	Windows 2000 Advanced Server SP4
	Windows 2003 Server (also install ASP and ASP.NET)
	Windows XP SP2
Web Server	IIS 5.0, 5.1, or 6.0
Browser	Internet Explorer 6.0 SP1
	Netscape Navigator 7.1 or higher

Table 2-3 Workstation Requirements – Report Viewing

WebXM Agent Service, Content Agent, Infrastructure Agent and Interaction Agent

Table 2-4 lists the minimum operating systems and browsers required for the systems on which you plan to install the agent services.

Software	Version
Operating System	Windows 2000 Professional SP4
	Windows 2000 Server SP4
	Windows 2000 Advanced Server SP4
	Windows 2003 Server (also install ASP and ASP.NET)
	Windows XP SP2
Browser	Internet Explorer 6.0 SP1

Table 2-4 WebXM Agents and Agent Service Requirements

WebXM Alert Service

Table 2-5 lists the minimum operating systems and browsers required for the systems on which you plan to install the alert services.

Software	Version
Operating System	Windows 2000 Professional SP4
	Windows 2000 Server SP4
	Windows 2000 Advanced Server SP4
	Windows 2003 Server (also install ASP and ASP.NET)
	Windows XP SP2
Browser	Internet Explorer 6.0 SP1

Table 2-5 WebXM Alert Service Requirements

Additional Software Requirements

The WebXM Server Setup requires the .NET Framework, v1.1 SP1, and will not install without it.

If you are using WebXM with an Oracle database, one of the following database clients is required on each machine on which you are installing WebXM Server components:

- Oracle Provider for OLE DB Release 9.2.0.4 for an Oracle 9i database
- Oracle Provider for OLE DB Release 10.1.0.2 for an Oracle 10g database

System Requirements — WebXM Database

You can configure WebXM to save data to a Microsoft SQL Server database or an Oracle database. Table 2-6 lists the minimum operating systems and database software required for the system you plan to use for the WebXM database.

More specific details about WebXM databases can be found in “Using a SQL Server Database with WebXM” on page 5-4 and “Using an Oracle Database with WebXM” on page 5-11.

Software	Version
Database	SQL Server 2000 SP3
	MSDE 2000 (SQL Server 2000 Desktop Engine) SP3
	Oracle 9i Release 2 or higher
	Oracle 10g Release 1 or higher
Operating System	For SQL Server or MSDE:
	Windows 2000 Professional SP4
	Windows 2000 Server SP4
	Windows 2000 Advanced Server SP4
	Windows 2003 Server
	For Oracle:
	Refer to the Oracle documentation

Table 2-6 WebXM Database Requirements

System Requirements – WebXM Desktop

Tables 2-7 and 2-8 provide a summary of the minimum hardware and software required to run WebXM Desktop.

Hardware Requirements

Table 2-7 lists the minimum requirements for the systems on which you plan to install WebXM Desktop.

Hardware	Recommended Requirement
Processor	Pentium Class PC (PIII - 800 MHz)
Memory	128 MB RAM
Disk Space	40 MB

Table 2-7 Hardware Requirements for WebXM Desktop



You can use WebXM Desktop to scan large sites and generate all the reports; there are no limits imposed. The only limitations are hardware-related: the amount of memory, disk space, and processing speed.

Operating System and Other Software

Table 2-8 lists the minimum operating systems and other software required for the systems on which you plan to install WebXM Desktop.

Software	Version
Operating System	Windows 2000 Professional SP4
	Windows 2000 Server SP4
	Windows 2000 Advanced Server SP4
	Windows 2003 Server
	Windows XP Professional SP2
Browser	Internet Explorer 6.0 SP1
WebXM Desktop Setup file	.NET Framework, v1.1 SP1

Table 2-8 Software Requirements for WebXM Desktop

User Account Information

During the various stages of the installation and configuration of WebXM, you will have to enter the following user account information:

Local System User Account

When you are installing WebXM, the Local System User Account used to log on to each individual machine WebXM must have Administrator rights.

Service Account

During the configuration of the WebXM components you install, you must enter service account information. This service account allows the agents to access the database server.



The Local System User account and the Service Account can be a single account, with the same User name and Password.

If you are using a **SQL Server database** with WebXM, the service account must have the permissions that allow you to create a database and tables, add users, and grant rights.

With a SQL Server database, you can use a single service account or multiple service accounts, depending on how you decide to install WebXM. For example, you can use one account ID and password to install the content agents and a different account ID and password to install the interaction agents. However, if you are using an **Oracle database**, the service account used to configure each of the components must be the same.



The WebXM service account does not have to be the same as the Oracle database user account. During WebXM configuration, you can enter the database user credentials, and then the service account credentials.

The service accounts used for the WebXM agents and the WebXM database should have passwords that do not expire. If, however, the passwords must change at regular intervals, you can re-run the Configuration Wizard on all the Agent Server systems and enter the new details. For information about the Configuration Wizard, see “Configuring the WebXM Server Components” on page 3-14.

When the installation and configuration is finished, the service account will be listed as System Administrator in the Administration Center of the WebXM Control Center. For more information about user roles and permissions, refer to the *WebXM Administration Guide* and the *WebXM User Guide*.

File and Folder Permissions

The service account must have the following permissions on *Drive:\YourInstallFolder\Watchfire\WebXM* and all of its subfolders:

- Read and Execute
- Write
- Delete
- Delete files and subfolders
- Create files and subfolders

These permissions enable the service account to write to WebXM’s log files. They also enable the scan agents to write temp files, without which WebXM scans would not function. The WebXM Configuration Wizard creates these permissions for you — do not change them.

With an **Oracle database**, the WebXM service account also needs to have read access to the `tnsnames.ora` file. For Oracle 9i, the file can be found on *Drive:\Program Files\Oracle\ora92\network\admin*, where the `ora92`

folder may be different on your installation. For Oracle 10g, the file can be found on *Drive:\Oracle\product\10.1.0\Client_1\network\admin*.

What Is Your Existing Environment?

In order to identify the steps required to install WebXM, you must understand your existing environment. The website structure, organizational structure, and physical environment all factor into your decisions about how to install WebXM.

This section poses a series of questions that will help you determine what preparations you must make for the WebXM installation.

Website Structure

How is your website constructed? Do you have a single web server or a number of web servers? Where are the web servers? Are they all close together, or physically separated (as in different buildings, in different countries)? These issues are important when determining how many servers you will use and where you should install the WebXM Server components.

For example, if you have a number of web servers, each dedicated for a specific group in your organization, you might consider a separate WebXM Server installation for each one. In this scenario, you could use a separate database for each installation, or use a single database to store and process data collected by scans of all the individual web servers.

Organizational Structure

You need to understand how the organization is structured; who will be creating and running jobs, and generating reports; who will be Report Consumers; who is developing the content and may require WebXM Desktop.

You also need to understand the corporate standards that are in place. Will you use WebXM to manage the website content and ensure that these standards are adhered to?

While this issue relates more directly to the options you will define for your jobs, the volume and complexity of these jobs will affect how WebXM is installed.

Physical Environment

What are the specifications of the servers allocated for the WebXM installation? Ideally, you want to make the most powerful systems you have available for WebXM. The systems that you do have available will determine how you install and configure each of the components.

The specifications of your network are essentially invisible to WebXM. However, some aspects can affect performance. Here are some factors to consider:

Transmission medium: The type of cabling you use can affect the speed at which jobs are performed.

Firewalls: If you will be scanning websites located outside your network, the existence of the firewall and how it is configured will affect the speed with which jobs are performed.



For infrastructure scan job IP address lookups to work, the computer on which the infrastructure agent resides requires access to port 43 of at least the IP address of the WHOIS server that corresponds to the region in which your host is located. If your host is in North America, that means the *whois.arin.net* server. If your host is anywhere else, that means *whois.arin.net* plus the WHOIS server for your part of the world, which is one of the following: *whois.ripe.net*, *whois.apnic.net*, or *whois.lacnic.net*.

Proxy Servers: If your network uses a proxy server to access the Internet, you must specify the proxy settings for the web browser used by the service account.

Single Server Versus Multiple Servers

The number of servers that you need, and the configuration you use, depend on how you plan to use WebXM. The optimum configuration for your situation depends on a number of factors, including the number of users, size of the site, types of jobs to be performed, and required reports. Every situation is different; however, there are a few basic configurations that you can consider.

WebXM on a Single Server

The simplest configuration is to install all the WebXM Server components, including the database, on a single machine; see Figure 2-1. This configuration is suitable for smaller organizations or workgroups.

You can choose to store the scan information in an MSDE database, a SQL Server database, or an Oracle database.

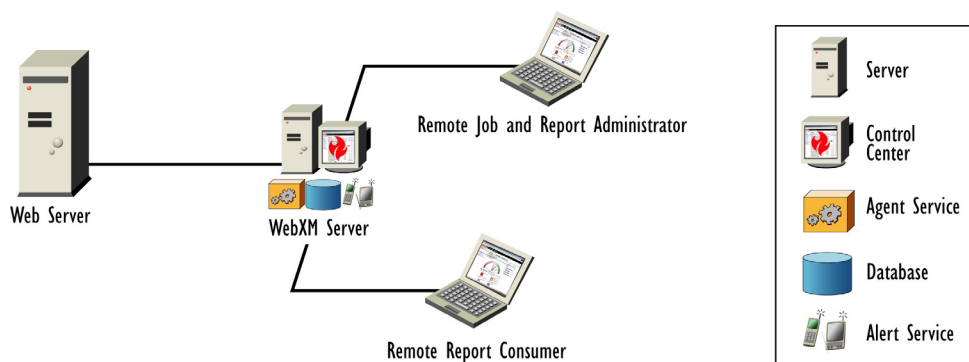


Figure 2-1 Single Server Configuration



This is the only WebXM configuration with which you can use MSDE, because MSDE does not meet the requirements of a multi-server environment.

For more information, see “Installing WebXM on a Single Server” on page 3-9.

WebXM on Multiple Servers

You may decide that a single server configuration is not sufficient to meet your requirements. If the number of scans you expect to run exceeds the performance limits of a single server, or if you want to use a dedicated SQL Server or Oracle database server, you can install the WebXM components across several machines. The following diagrams provide examples of WebXM components installed on more than one server.

Multiple Servers — Configuration A

In this scenario, as illustrated in Figure 2-2, the WebXM Control Center, alerting service, agent service, and the SQL Server or Oracle database are installed on a single machine. Additional agents are installed on separate servers.

The number of agent servers you install depends on the volume of scans you expect to perform, and their relative complexity.

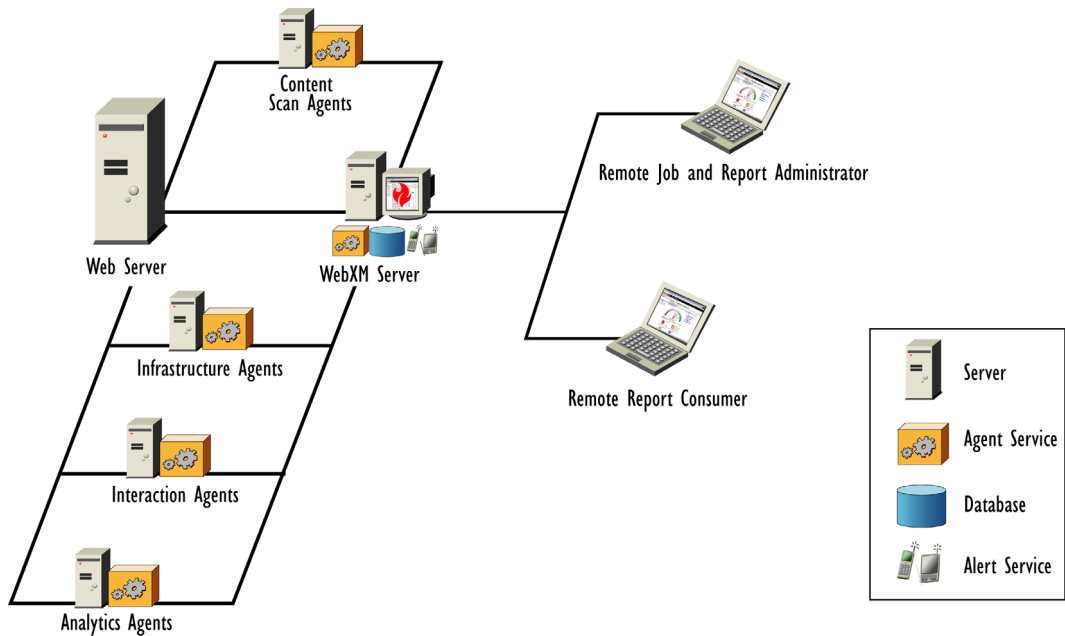


Figure 2-2 Multiple Servers — Configuration A

Multiple Servers — Configuration B

In this scenario, as illustrated in Figure 2-3, the WebXM Control Center, alerting service, and a few agents are installed on one server. The SQL Server or Oracle database resides on a separate server, and additional agent servers have been installed.

Installing the WebXM Control Center and the database on separate servers can improve overall performance since you do not have a single server performing the scan and storing the data at one time.

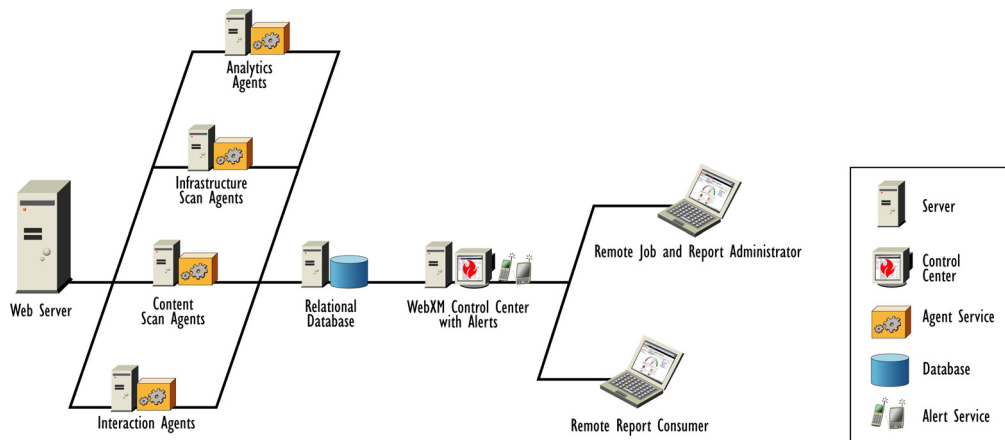


Figure 2-3 Multiple Servers — Configuration B

Multiple Instances of WebXM on a Single Server

When installing and configuring WebXM, you can choose to configure the default database, or create a number of separate installations on a single server. Each installation, or instance, has its own database and corresponding dashboard. You can install multiple instances of the WebXM Control Center or agents.

Multiple Instances of the WebXM Control Center

In certain situations, you may want to consider installing a number of WebXM Control Centers on a single machine. Each installation, or instance, is independent of the others, with its own virtual directory and database access.

For example, if your organization includes separate business units, each unit could have its own WebXM Control Center at a different virtual directory, accessing its own dedicated database. Figure 2-4 illustrates this example.

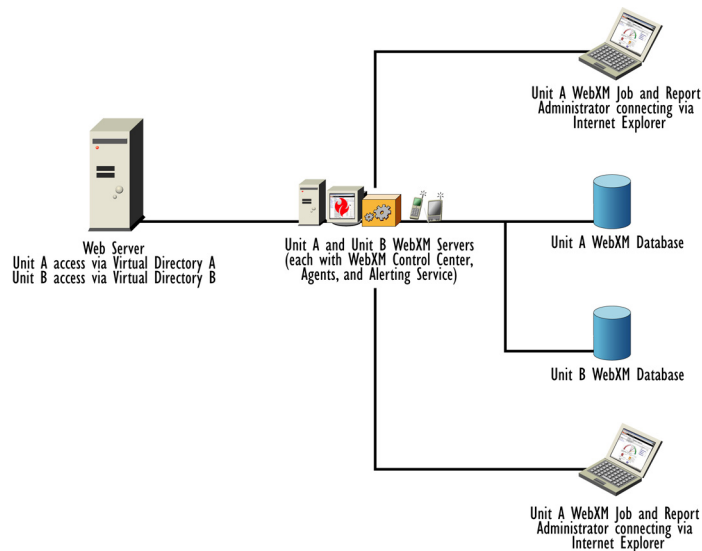


Figure 2-4 Multiple Instances of the WebXM Control Center on one Server

Multiple WebXM Control Centers on a single server could also be used in the following scenarios:

- You want to set up a demonstration WebXM environment that will not interfere with the production WebXM environment
- You can have one WebXM Control Center for scans of your Intranet sites and one for scans of your Internet sites

Multiple Instances of WebXM Agents

You can also install a number of instances of the agents on a single server, as illustrated in Figure 2-5. This allows you to run a number of different scan jobs, each accessing a different database.

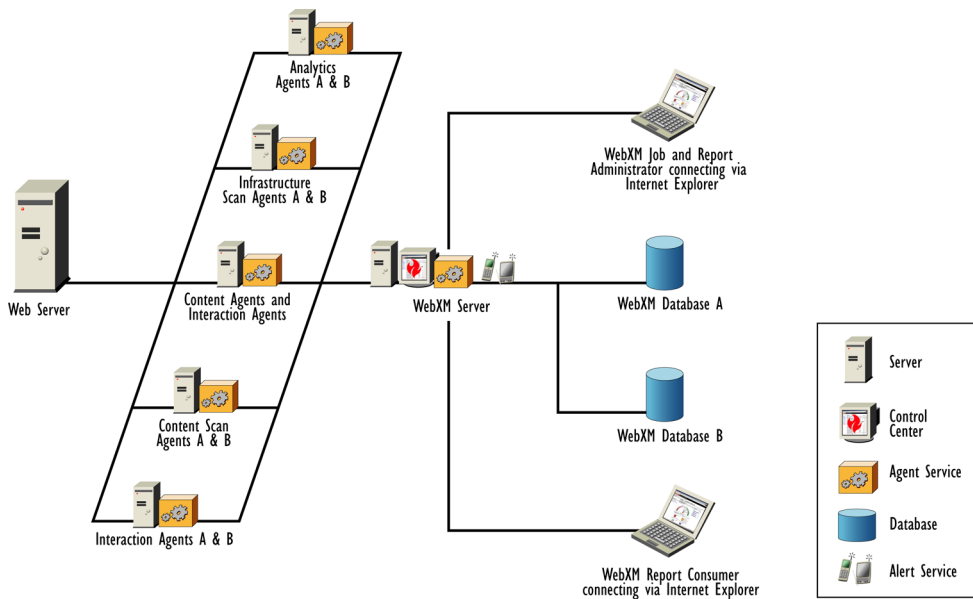


Figure 2-5 Multiple Instances of the Agents on one Server

Having a single server with agents serving different WebXM databases allows a large organization, that has more than one WebXM Control Center, to more efficiently allocate agent servers and better manage agent server capacity, potentially saving hardware cost.

Installing multiple agents on the same physical server can minimize the “quiet” time on an agent server, which occurs when a given WebXM Control Center has no scan jobs running while another may be running several jobs.

For installation and configuration instructions, see “Installing Multiple Instances of WebXM on a Single Server” on page 3-13.

INSTALLING WEBXM

Topics

- Install Summary
- Preinstall Checklist
- Preparing for Installation
- Installing WebXM Server
- Configuring the WebXM Server Components
- Activating the Optional Modules
- Configuring NetGenesis Database for Traffic
- Configuring NetGenesis Database for Traffic
- Installing MSDE
- Installing WebXM Desktop
- Upgrading WebXM — A Checklist

Install Summary

After you have analyzed your requirements and planned the deployment of WebXM, you are ready to begin the installation. Table 3-1 provides a typical WebXM installation workflow.

Install Step		For More Information
1	Prepare the system for installation	“Preinstall Checklist” on page 3-2 and “Preparing for Installation” on page 3-6
2	Prepare the database server	“Using a SQL Server Database with WebXM” on page 5-4 “Using an Oracle Database with WebXM” on page 5-11
3	Install WebXM Server	“Installing WebXM on a Single Server” on page 3-9, or “Installing WebXM on Multiple Servers” on page 3-10
4	Configure the WebXM Server components	“Configuring the WebXM Server Components” on page 3-14
5	Install Traffic (optional)	“Configuring NetGenesis Database for Traffic” on page 3-18
6	Install WebXM Desktop (optional)	“Installing WebXM Desktop” on page 3-20

Table 3-1 A Typical Installation Workflow

Preinstall Checklist

Table 3-2 describes the steps you need to take before installing WebXM. Consult the ReadMeServer file on your installation CD for up-to-date system requirements and for the correct version of any required software.

✓ Preinstall Step
HARDWARE
1 Ensure that hardware is in place for the following components: WebXM Control Center server, Agent server, and Database server.
2 Plan your installation environment so that it meets your needs; see “What Is Your Existing Environment?” on page 2-12 and “Single Server Versus Multiple Servers” on page 2-13. For hardware recommendations, consult your Account Manager or Professional Services.
SOFTWARE

Table 3-2 Preinstall Checklist

✓ Preinstall Step	
3	The WebXM Server Setup requires the .NET Framework, v1.1 SP1, and will not install without it. The .NET Framework can be obtained from Microsoft's website.
4	Verify that you are installing the correct version of WebXM.
5	Verify that you have a valid license for WebXM and its modules (e.g., Quality, Privacy), and for WebXM Desktop.
6	Ensure that the latest critical updates for Microsoft Windows Operating System are installed on all servers with WebXM components.
7	WebXM Control Center: Ensure that you have the correct version of the Windows operating system, IIS, and Internet Explorer for the WebXM Control Center server.
8	Agent Server or Servers: Ensure that you have the correct version of the Windows Operating system, Internet Explorer, and Flash (if WebXM will be scanning Flash media) for each Agent server. When scanning websites in other languages, ensure that you install all language packs on each content Agent Server(s); see "Installing Language Packs for International Website Scanning" on page 3-7.
9	Database: Ensure that you have the correct version of SQL Server or Oracle.
10	Database: If you are using SQL Server, ensure that you use the appropriate version for the amount of pages being scanned; if you have over 5,000 pages, SQL Server Enterprise is recommended.
11	Individual Workstations: Ensure that those accessing reports through their workstations have the correct version of the Windows Operation System, Internet Explorer, and/or Netscape Navigator.
12	Windows Server 2003: Ensure that ASP and ASP.net are installed.
USER ACCOUNTS	
There are two user accounts required to install and run WebXM: Installation Account and Service Account.	
13	Installation Account: Ensure that the Installation Account is an account with local administrator rights. If the Service Account has administrator rights, then it can also be used for the Installation Account.
14	Service Account: Ensure that the Service Account is a dedicated account with a password that does not expire. If the password must be changed for security reasons, the WebXM configuration wizard will have to run after the password has changed.
15	Service Account: In a multiple WebXM server installation, ensure that the Service Account is a domain account. In a single WebXM server installation, the Service Account can be a local account.
16	Service Account: Ensure that the Service Account is an administrator so that all the user permissions are in place. If there are internal security reasons why the Service Account cannot be an administrator, the following permissions are required for the install location, drive:\ \ InstallFolder\Watchfire\WebXM, and all its subfolders: <ul style="list-style-type: none"> - Read and Execute - Write - Delete - Delete files and subfolders - Create files and subfolders
DATABASE: GENERAL	

Table 3-2 Preinstall Checklist (continued)

✓ Preinstall Step	
17	Ensure that the storage device has sufficient space to store data. The equation for determining the size you need is approximately 200 kB times the maximum number of pages you are scanning. The 200 is a derivative of 20 kB times 10 unique links per page.
18	Ensure that your version of SQL or Oracle is supported.
19	Ensure that you have a database backup plan.
DATABASE: SQL SERVER	
20	Ensure that the WebXM Service Account has permissions to create a database, add tables, add users, and grant rights. Normally these permissions equate to System Administrator rights on the SQL Server database. If there are security reasons why the Service Account cannot be a System Administrator, it can be downgraded to DB Owner of the WebXM database after the installation.
21	You can improve SQL Server performance by changing some of its settings; see “SQL Server Configuration” on page 5-5.
DATABASE: ORACLE	
22	Confirm that you have the following information: host string of the database, user name, and password of the User Account.
23	Ensure that Oracle Client is installed on each of the WebXM servers.
24	Ensure that the Oracle Provider for OLE DB is installed.
25	Create a tablespace on the database server with an initial space allocation of 32 kB. The tablespace can be any name, but WebXM is usually used.
26	Ensure that the Registry entry for HKLM\SOFTWARE\ORACLE\ORACLE_HOME, points to the correct location of your Oracle installation. Oracle software upgrades may cause this entry to be incorrect.
27	Using the Oracle Net Configuration Assistant, create a listener on the Oracle database server.
28	Ensure that the option Locally managed tablespaces is enabled.
29	Ensure that a User Account is created with the following: <ul style="list-style-type: none"> - Standard roles Connect and Resource - Permissions, which need to be created explicitly and not defined as a role: Alter Any Table (for upgrade only), Create Table, Create Procedure, and Create Sequence
30	To save space on the database, turn off enable logging.
31	The tnsnames.ora file on the WebXM server is used to resolve the connection between the WebXM server and the Oracle database. If the tnsnames.ora is not present, run the Oracle Net Configuration Assistant on the WebXM Server(s) to create the tnsnames.ora file and connection.
32	Using the query <code>SELECT * FROM NLS_DATABASE_PARAMETERS;</code> ensure that the character sets and language settings are as follows: <ul style="list-style-type: none"> - NLS_CHARACTERSET: AL32UTF8 - NLS_NCHAR_CHARACTERSET: AL16UTF16
DATABASE: MSDE	

Table 3-2 Preinstall Checklist (continued)

<div> <div>✓</div> <div>Preinstall Step</div> </div>
<div> <div>33</div> <div> <div>After installing MSDE, perform the following to ensure that the WebXM Configuration Wizard can connect to it:</div> <ul style="list-style-type: none"> - Open the Services Panel in MSDE and set the SQL Server Agent to automatic. - Start the Services prior to installing WebXM. </div> </div>
<div> <div>NETWORK</div> </div>
<div> <div>34</div> <div> <div>Identify any proxy servers or firewalls between the following connections:</div> <ul style="list-style-type: none"> - WebXM Agent Server(s) and the websites being scanned - Individual workstations and the WebXM Control Center Server - WebXM Control Center Server and the websites being scanned <div>Note that the less proxy servers and firewalls between these connections, the better the performance of WebXM.</div> </div> </div>
<div> <div>35</div> <div> <div>To configure email alerts the alert agent needs access to the Mail Server. Confirm the following for the WebXM Server hosting the alert agent:</div> <ul style="list-style-type: none"> - The WebXM Server has a connection with the Mail Server - The host name of the Mail Server </div> </div>
<div> <div>36</div> <div> <div>If you are installing the Inventory module, one of the reports requires port 43 to be open on those WebXM servers that are running its agents. The purpose of the open port is to obtain WHOIS information on the servers being scanned.</div> </div> </div>
<div> <div>37</div> <div> <div>WebXM uses NT authentication for user access. Determine whether or not the WebXM users are trusted on the Control Center server. If not, the users can be added locally, although NT authenticated users are easier to maintain.</div> </div> </div>
<div> <div>38</div> <div> <div>If SiteMinder is being used in your environment for user access, consult Professional Services about customizing WebXM for user access using SiteMinder.</div> </div> </div>

Table 3-2 Preinstall Checklist (continued)

✓ Preinstall Step

PRE-DEPLOYMENT CONSULTANT LOGISTICS

39 Ensure the following resources are available to your consultant over the course of the deployment:

- Business resource — normally the project manager for the WebXM deployment
- IT resource
- Network resource
- Database resource
- CMS administrator
- CMS (workflow) developer

The latter two resources would only be needed in a CMS integration.

40 Ensure the consultant has their own access, either directly or remotely, to the following servers: WebXM Control Center, Agent(s), Database, and CMS Server (if applicable). If access cannot be granted to the Database server, which normally occurs in cases with Oracle installs, make sure the Database resource is easily available to help the consultant with any information needed from the Database server.

Table 3-2 Preinstall Checklist (continued)

Preparing for Installation

Before you begin installing WebXM, ensure that the equipment is ready and that you have all the required information available.

Information to Have on Hand

The following information should be available at the time of installation:

- If you are using a Microsoft SQL Server database:
 - Name of the machine designated as the WebXM database server
 - Name of the WebXM database

This can be the name you want to give a new database or the name of an existing database
- If you are using an Oracle database:
 - Host string of the database

The database must be created before you install WebXM

- User name and password used to create the Oracle database
- License for WebXM Server

This license includes activation for any optional modules or plug-ins you purchased, as well as scanning page limit information
- Service account user name and password for the database server and agent servers
- License Key for WebXM Desktop

System Requirements

Ensure that the systems on which you intend to install WebXM meet or exceed the minimum requirements, as described in “System Requirements — WebXM Server” on page 2-5 and “System Requirements — WebXM Desktop” on page 2-9.

Preparing for WebXM Server Installation

You can prepare each system for the WebXM Server installation by addressing the following issues:

- If you have other versions of Watchfire software on your systems, such as WebXM Desktop, remove all those applications
- The account you use to log on to the machine on which you are installing WebXM Server must have local Administrator rights
- Ensure that no one is accessing the WebXM database

For WebXM Server installation instructions, see “Installing WebXM Server” on page 3-8.

Installing Language Packs for International Website Scanning

When scanning a website in another language using a custom text rule or a metatag content rule, ensure that each server hosting the content scan agent has all language packs installed on it. If the scan agent cannot understand the content of the pages because the operating system does not have the proper language installed, then it cannot properly find your text or metatag content rules on the website.

To install language packs for Windows 2000 and XP operating systems:

- 1 Go to the Control Panel and click **Regional Options**.
- 2 On the General Tab, select all the languages not currently installed and click **OK**. The operating system will prompt you for the installation CD for Windows.
- 3 Enter the path to the installation CD and click **OK**. All language packs will be installed by Windows. It may be necessary to re-install Windows service packs if they are not already on the installation CD.

Preparing for WebXM Desktop Installation

You can prepare each system for WebXM Desktop by addressing the following issues:

- If you have other versions of Watchfire software on your systems, remove those applications
- Choose an account with local Administrator rights on the local machine

For installation instructions, see “Installing WebXM Desktop” on page 3-20.

Installing WebXM Server

This section takes you through the steps to install WebXM Server.

You can choose to install all the WebXM Server components on a single server, or install the components on separate servers, depending on the configuration you defined during the planning stage.

The agents for optional modules and plug-ins are automatically installed when you install WebXM. They are activated when you have a valid license.

Installing WebXM on a Single Server

Follow this procedure if you want to install all the WebXM Server components on a single machine.



If you decide to use WebXM Server with an MSDE database, you must install MSDE first. For more information, see “Installing MSDE” on page 3-19.



Any technologies that you use on your website must also be installed on the WebXM Server. For example, if you use Flash on any web pages, you must have the correct version of Flash installed.

To install and configure WebXM on a single server:

- 1 Insert the WebXM CD in the CD-ROM drive.
 - If you have autorun enabled, the WebXM Welcome screen opens, displaying the Installation and Configuration Menu. Click **Install WebXM Server**
 - If autorun is not enabled, or the Welcome screen does not open, navigate to the CD-ROM drive and double-click the file `WebXM_ServerSetup.exe`
- 2 In the Watchfire Setup Wizard, click **Next**.
- 3 In the License Agreement dialog box, choose the **I accept the terms in the license agreement** option, and then click **Next**.
- 4 In the Customer Information dialog box, enter your **User Name** and **Organization**, and then click **Next**.
- 5 In the Setup Type dialog box in the wizard, choose the **Complete** setup option to install all the WebXM components on the single server, and then click **Next**.
- 6 In the Destination Folder dialog box, click **Next** to accept the default installation locations.

OR

Click **Change** to choose different installation locations, and then click **Next**.

- 7 In the Ready to Install dialog box, click **Install** to proceed with the installation.

WebXM Server is installed in the location you specified. When the installation is finished, the Setup Wizard Completed dialog box opens.

- 8 At this point, you can proceed with the WebXM Server configuration or exit the installation.
 - If you want to configure WebXM Server, ensure the **Launch Configuration Wizard** check box is selected, and then click **Finish**
 - If you want to end here and perform the configuration later, clear the **Launch Configuration Wizard** check box, and then click **Finish**

For more information about the configuration process, see “Configuring the WebXM Server Components” on page 3-14.

Installing WebXM on Multiple Servers

Follow this procedure if you want to install WebXM Server components on separate machines. The following components can be installed on individual servers, alone or in combination:

- WebXM Control Center
- Content, Infrastructure, and Interaction Agents
- Collection and Processing Agents (Analytics module)
- Alerting Service

For example, you might install the Control Center and alerting services on one machine, the content agents on a second machine, and the interaction agents on yet another machine. For sample configurations, see “WebXM on Multiple Servers” on page 2-14.

Follow this procedure for each machine on which you are installing a component of WebXM Server.



WebXM components must be installed locally. You cannot install the WebXM Control Center on one computer, and then install the other components over the network.

The servers on which you install the agents must also have any technologies installed that you use in your web pages. For example, if you use Flash on your website, you must have the correct version of Flash installed.

To install WebXM on multiple servers:

- 1 Insert the WebXM CD in the CD-ROM drive.
 - If you have autorun enabled, the WebXM Welcome screen opens, displaying the Installation and Configuration Menu. Click **Install WebXM Server**
 - If autorun is not enabled, or the Welcome screen does not open, navigate to the CD-ROM drive and double-click the file `WebXM_ServerSetup.exe`
- 2 In the menu, under Watchfire WebXM Server, select **Install**.
- 3 In the File Download dialog box, click **Open** to start the installation. The Watchfire Setup Wizard opens to the first dialog box.
- 4 Click **Next** and follow the instructions to start the installation.
- 5 When you reach the Setup Type dialog box in the wizard, select the **Custom** setup option, and then select the components that you want to install on the local machine. You can also check the available disk space by clicking **Space**.

WebXM Server is installed in the location you specified. When the installation is finished, the Setup Wizard Completed dialog box opens.

- 6 At this point, you can proceed with the WebXM Server configuration or exit the installation.
 - If you want to configure WebXM Server, ensure the **Launch Configuration Wizard** check box is selected, and then click **Finish**
 - If you want to end here and perform the configuration later, clear the **Launch Configuration Wizard** check box, and then click **Finish**

For more information about the configuration process, see “Configuring the WebXM Server Components” on page 3-14.

Installing Additional Agents

If, at some point after the initial WebXM installation, you determine that you do not have enough agents available to perform the required scans, you can install additional agent services.

Ideally, these agents should be installed as close as possible to the web servers they are scanning. By reducing the number of hops between the agent and the web server, you can improve the speed of the scan.

To install additional agents, follow the steps in “Installing WebXM on Multiple Servers” on page 3-10.

Why Would I Want to Install Additional Agents?

When a job is created, it is added to the queue where an agent picks it up and runs it. If there is no agent to run the job when it is due, it will be picked up as soon as an agent becomes available.

If jobs are continually not picked up on schedule, it could be an indication that more agents are required.

Installing Multiple Instances of WebXM on a Single Server

When you install WebXM, you have the option of installing a default database, or multiple instances of the database on a single machine. Each instance is an independent set of WebXM configuration information, with its own database and corresponding dashboard.

You may want to consider installing multiple instances of WebXM when you need to support multiple environments on a single large server. For example, if your organization is structured into business units, each with its own website, you can install one instance of WebXM Server for each. This allows each group to have its own WebXM Control Center and database, independent of the others, with a dashboard that represents the health of the unit's website.

You can install multiple instances of the WebXM Control Center or the Agents. For installation instructions, see “Configuring the WebXM Server Components” on page 3-14.

You may want to install the default instance first, and then install additional named instances as required. There is no limit to the number of named instances that you can run on a single computer.



If you add a new instance using the WebXM Configuration Wizard, you must restart the Agent Service in order to incorporate the change.



If you are installing more than one instance of WebXM on a single machine, you may want to consider scaling the hardware accordingly. Running a number of instances of WebXM should not adversely affect the performance of the scan; however, how your web server is structured, the kinds of technologies used on the website, etc., can affect the amount of resources required by each instance.

Configuring the WebXM Server Components

After you install or upgrade WebXM Server, either on a single machine or across multiple systems, you must configure each installed component. Therefore, the Configuration Wizard must be run on all instances of WebXM and on all servers.

During configuration you define the name and location of the WebXM database to be used, as well as the WebXM service account name and password.

To configure WebXM Server components:

- 1 Launch the WebXM Configuration Wizard.
 - After installing WebXM Server, select the **Launch Configuration Wizard** check box in the Setup Wizard Completed dialog box, or
 - On the Windows Start menu, select **Programs > Watchfire > WebXM Configuration Wizard**The Welcome screen opens.
- 2 In the Welcome screen, click **Next**.
The Instance Name dialog box opens.
- 3 Specify the name of the instance you want to configure.
 - If you are installing only one instance of WebXM on this machine, select the **Select or create a default instance** check box
 - If you are installing more than one instance of WebXM on this machine, clear the **Select or create a default instance** check box, enter a name for the instance, and then click **Next**. You will be given the option to configure another instance at the end of the wizard
- 4 In the Database Type dialog box, select the type of database you will use with WebXM, and then click **Next**.
 - If you selected Microsoft SQL Server, the Database Connection dialog box opens. Proceed to Step 5
 - If you selected Oracle, the Database Connection dialog box opens. Proceed to Step 7
- 5 (SQL Server only) In the Database Connection dialog box, enter the name of the database server, or select one from the **SQL Server or Server\Instance name** drop-down list.

- 6 (SQL Server only) Enter the name of the database, or select one from the **Database Name** drop-down list, and then click **Next**.
 - If you are using an existing database, the License dialog box opens. Proceed to Step 9



If the version of the existing database you identify is incompatible with WebXM, a message is displayed, warning you that all data will be erased. If you click **Yes** to continue, a second message box asks you to confirm that you want to erase the data in the selected database.

- If you are creating a new database, a message box opens asking for confirmation that you want to create the database. Click **Yes** to continue. The License dialog box opens. Proceed to Step 9
- 7 (Oracle only) Enter the **User name** and **Password** you specified when you created the Oracle database.
 - 8 (Oracle only) Enter the **Host string** of the new Oracle database, and then click **Next**.



If Oracle Provider is not installed, a message is displayed, instructing you to install Oracle Provider for OLE DB Release 9.2.0.4. If the installed version of Oracle Provider is not Release 9.2.0.4, a message is displayed, instructing you to install the required version. If your database is Oracle 10g, you should install the Oracle Provider for OLE DB Release 10.1.0.2.

- 9 In the License dialog box click Load License File and find the WebXM Server license file on your computer. Once the file is selected, your license information appears in the dialog box. Click **Next**.

The Service Account dialog box opens.

- 10 Enter the **Domain\user name** and **Password**, and then click **Next**.

The Authentication Type dialog box opens.

You can also browse the list of service accounts and select the one you want to enter.

- 11 Select the type of user authentication that the WebXM Control Center will accept:
 - If your organization presently uses Windows as a means of user authentication, choose **Integrated Windows authentication**. The System Administrator dialog box opens. Proceed to Step 13

- If your organization presently uses SiteMinder authentication, choose **Netegrity SiteMinder authentication**. The SiteMinder Authentication dialog box opens. Proceed to Step 12
- 12** (SiteMinder only) In the SiteMinder Authentication dialog box, select **pre-authorized authentication**. The System Administrator dialog box opens. For more details about this method, see “Pre-authorized Authentication” on page 3-22
- 13** Enter the user name and full name of the System Administrator. In a SiteMinder installation, the user name must be entered exactly as it appears in SiteMinder.

If the Analytics module is licensed, the Collection Data Folder dialog box opens. Proceed to the next step.

If the Analytics module is not licensed, the Specifications Complete dialog box opens. Proceed to step 16.
- 14** (Analytics module only) Specify the data folder, or use the default folder provided, where analytics data will be temporarily stored. This folder will be used by the collection agent to store visitor data as it is collected from your website by the data tags. If you have installed multiple collection agents, enter their folders in the next step. For details about how visitor data is collected and stored, see “Architecture of an Analytics Deployment” on page 4-2.

The Session Data Folders dialog box opens.
- 15** (Analytics module only) Enter any additional folders that are storing collected visitor data. When you have deployed multiple collection agents, each agent stores its data in its own processing data folder.
- 16** Click **Finish**. The Specifications Complete dialog box opens.



Before you continue, ensure that nobody is accessing the WebXM database.

- 17** (optional) Select the **Start the WebXM Service** check box to automatically start the service.



If you do not choose to automatically start the service, any jobs created by users will not be picked up by the agents. You can manually start the service using the Administrative Tools; see “Verifying the Agent Service and Alerting Service Installation” on page 6-4.

- 18 (optional) Select the **Launch the WebXM Control Center** check box to open WebXM in your browser.
- 19 (optional) Select the **Restart the Configuration Wizard...** if you want to configure another instance of WebXM.
- 20 Click **Exit**.

WebXM Server is now configured and connected to the database you specified.

Modifying the Configuration of WebXM Server

Using the Watchfire Configuration Manager, you can make changes to the WebXM configuration at any time. Such changes can include:

- Connecting to a different database
- Creating a new database
- Selecting a new license file
- Configuring additional WebXM components, such as content agents

To modify your WebXM configuration, follow the steps in “Configuring the WebXM Server Components” on page 3-14, making the desired changes.

Activating the Optional Modules

If you purchased an optional module or plug-in with WebXM Server, the components are automatically installed during the initial installation and the data tables are created in the WebXM database.

If you purchase a new module at a later date, you can activate it by selecting the new license file in the WebXM Control Center.



If you are changing a license which causes you to have less modules, WebXM will delete any jobs and reports that do not apply to your installation. For example, if you remove the Quality module, WebXM will delete any interaction test jobs and reports that you have created. In this case, you may wish to download your interaction jobs to WebXM Desktop before you change the license, otherwise they will be lost.

If an unlicensed job cannot be deleted by WebXM, you must delete it manually. You cannot view or run an unlicensed job.

To activate a new module:

- 1 In Internet Explorer, open the WebXM Control Center.
- 2 In the navigation bar, click **Administration Center**.
- 3 On the General tab, click **Manage Licensing**.

The License Management page opens, displaying a summary of your current licenses.

- 4 Click **Change License Information....**

The Change License Information page opens.

- 5 Enter the path to your new license file or click **Browse** to find it on your system.

Traffic integration and privacy management are configured for each separate content scan in the Report Data Collection properties. For more information, refer to the *WebXM User Guide*.

Configuring NetGenesis Database for Traffic

Before you can integrate traffic data with a content scan job, you need to run a SQL script on the NetGenesis database to prepare it for integration. The SQL script is found in your WebXM installation folder.

To prepare the NetGenesis database for integration with WebXM:

- 1 Open Query Analyzer.
- 2 Connect to the NetGenesis database.

- 3 Open the SQL script for your version of NetGenesis, found at *<WebXM Install Directory>\NetGen*, as follows:
 - MSNetGen.sql which works with NetGenesis Server 5.01 and 5.02
 - MSNetGen55.sql which works with NetGenesis Server 5.5
- 4 Run the query appropriate to your version of NetGenesis.

Installing MSDE

If you want to install all the WebXM components on a single server and store the scan data in an MSDE database, you must first install the MSDE software. It is available on the WebXM CD.



Please read the MSDE release notes and readme files for information about the installation and the **sa** password set up.

To install MSDE:

- 1 Insert the WebXM CD in the CD-ROM drive.
 - If you have autorun enabled, the WebXM Welcome screen opens, displaying the Installation and Configuration Menu. Click **Install MSDE 2000**
 - If autorun is not enabled, or the Welcome screen does not open, navigate to the MSDE2000 folder on the Watchfire WebXM CD, and then double-click the file Setup.exe
- 2 In the menu, under Watchfire WebXM Server, select **Install**.

Installing WebXM Desktop

There are two methods of installing WebXM Desktop on the individual development workstations:

- Install from the WebXM CD
- Download the executable from the WebXM Control Center

Typically, you would install WebXM Desktop from the CD during the initial WebXM installation process. WebXM Server allows you to download a copy of the set up file for WebXM Desktop, which you can run on a local machine, as long as you have the product license key. You can use the download method to install WebXM Desktop on additional workstations any time after the initial installation.

When you install WebXM Server, the file *WebXM_DesktopSetup.exe* is saved to */Program Files/Watchfire/WebXM/Web.App/downloads*. After the WebXM Control Center is installed, users can download a copy of this file to their workstations, and then install WebXM Desktop.

To install WebXM Desktop:

- 1 Insert the WebXM CD in the CD-ROM drive.
 - If you have autorun enabled, the WebXM Welcome screen opens, displaying the Installation and Configuration Menu
 - If autorun is not enabled, or the Welcome screen does not open, navigate to the CD-ROM drive and double click the file *WebXM_DesktopSetup.exe*
- 2 On the Welcome screen, click **Next**.

The License Agreement dialog box opens.
- 3 Choose the **I accept...** option, and then click **Next**.

The Customer Information dialog box opens.
- 4 Enter your User Name and Organization, select an installation option, and then click **Next**.

The Destination Folder dialog box opens.

- 5 The default destination folder is displayed. To accept the default and continue with the installation, click **Next**.
- If you want to change the destination, click **Change** and select the new folder.
- The Ready to Install dialog box opens.
- 6 Click **Install**.
- The WebXM Desktop applications are installed in the location you specified. When the installation is finished, the Setup Wizard Completed dialog box opens.
- 7 Click **Finish**.

Upgrading WebXM – A Checklist

Table 3-3 provides a typical WebXM upgrade workflow, intended to guide you through the process of upgrading an existing WebXM installation.

For tips on upgrading WebXM, see Chapter 6, “Tips, Tricks, and Troubleshooting”.

Step	Procedure	For More Information
1	Upgrade the database software, if required	“System Requirements — WebXM Database” on page 2-8
3	Install the WebXM Server upgrade	“Installing WebXM on a Single Server” on page 3-9, or “Installing WebXM on Multiple Servers” on page 3-10
4	Configure the WebXM Server components	“Configuring the WebXM Server Components” on page 3-14
5	Install WebXM Desktop upgrade	“Installing WebXM Desktop” on page 3-20
6	Test the new installations to ensure that the scan options you used in the previous version upgraded correctly.	

Table 3-3 A Typical WebXM Upgrade Workflow

SiteMinder and WebXM Integration

The integration of WebXM with Netegrity SiteMinder allows organizations to continue using SiteMinder as a means of centralizing user authentication and authorization.

With this integration, when a user navigates to WebXM they are prompted for their SiteMinder user name and password. It is then up to SiteMinder to authenticate the user and authorize them to access WebXM. If the user is not authenticated or authorized to use WebXM, they are sent to a log-in failed page. When SiteMinder is satisfied, WebXM determines the user's validity and access rights.



The integration of WebXM with SiteMinder does not extend to scanning websites that are protected with SiteMinder.

Pre-authorized Authentication

WebXM receives the user name of an authenticated and authorized SiteMinder user from the standard SiteMinder HTTP_SM_USER header, or HTTP_SMUSER header, which is set by the SiteMinder web agent. WebXM then determines the access rights of the user.

WebXM Desktop Support

It is possible to open a job from a webspace using WebXM Desktop when you have integrated WebXM with SiteMinder. There are some requirements that must be entered using the Desktop Options menu command:

- You need to enter a fully qualified domain name for the WebXM Server, such as *http://WatchfireInternal/webxm/*
- You also need to enter the path and name of the *.fcc file used by SiteMinder, as well as a SiteMinder user name and password

IMPLEMENTING THE ANALYTICS MODULE

Topics

- Introduction
- Architecture of an Analytics Deployment
- Scaling an Analytics Deployment
- Data Tag Syntax and Examples
- Methods for Populating a Website With Data Tags
- Deploying Analytics on Your Website: a Summary

Introduction

The Analytics module of WebXM shows you how visitors interact with website content and measures the effectiveness of that content. It is an optional module that integrates with all WebXM modules. Since it is optional, a valid license key is required to enable Analytics during installation of WebXM. For more details about modules and their activation, see “Activating the Optional Modules” on page 3-17 and refer to the *WebXM User Guide*.

Unlike other modules of WebXM, Analytics uses a data tag inserted into your website pages to collect data about visitors. The data tag is a combination of HTML and JavaScript which together comprise a few lines of code. Parameters within the HTML control the type of information collected as well as how a particular page is represented within WebXM’s reports. Before information can be collected from your visitors, their browsers must have JavaScript enabled.

Before implementing the Analytics module, please review the contents of this chapter. It is important that you plan the implementation carefully so as to yield the following results:

- The visitor data that WebXM collects is useful to everyone who needs it
- The visitor data is collected from those areas of your website or web properties that are most important to your organization
- The visitor data is displayed clearly and logically on the My Watchfire dashboard and within WebXM’s reports

Once you have read this chapter and understood all the implications of an implementation, it is important to refer to “Deploying Analytics on Your Website: a Summary” on page 4-20 for a description on how to proceed next.

Architecture of an Analytics Deployment

The Analytics module collects visitor information from a website using a web application called a collection agent. This agent relies on a combination of JavaScript and HTML code — called a data tag — that is inserted into each web page being tracked. The collection agent gathers data about visitors as they navigate through your website, visiting HTML pages with an inserted data tag.

The entire data collection process followed by Analytics is shown in Figures 4-1 and 4-2 and summarized as follows:

- 1** The browser requests a page from the web server.
- 2** The web server returns the page plus the data tag reference.
- 3** The browser requests the data tag from the WebXM server. The collection agent is responsible for serving the JavaScript to the browser.
- 4** The browser executes the data tag code.
- 5** The data tag code gathers visitor information required by the WebXM server, such as browser information, page views, and time and hour of visits. Note that WebXM does not collect or store sensitive information about visitors.
- 6** The data tag code requests an image from the WebXM server, with the visitor information being carried back on top of the request.
- 7** The collection agent collects the detected values and returns an invisible 1x1 GIF image plus cookies. The image is loaded in the background and has no visible affect on the page. In fact, the data collection itself is independent of the web server that is being tracked.

The detection of metrics is very fast, typically less than 100 milliseconds, and imperceptible to the visitor.
- 8** The collection agent stores the data tag information it has collected in a data folder. Each collection agent, if you have installed more than one, has its own data folder in which to store the information collected. When you have multiple data folders, sometimes the data from the same user session can reside in different data folders.
- 9** The session agent gathers visitor information from all the collection agents and assembles it into user sessions.
- 10** The session agent stores the partial and completed sessions in the WebXM database.
- 11** The processing agent retrieves the completed user sessions, analyzes them, and summarizes them into something meaningful that can then be displayed in WebXM's reports. The processed information is stored, as analytics data, in the WebXM database.
- 12** An analytics job is configured to display the collected data from certain regions, or domains, of your website. It is also configured to display certain kinds of data, such as the number of page views, in the form of reports.



Visitor data cannot begin to be collected from your website until an analytics job is created.

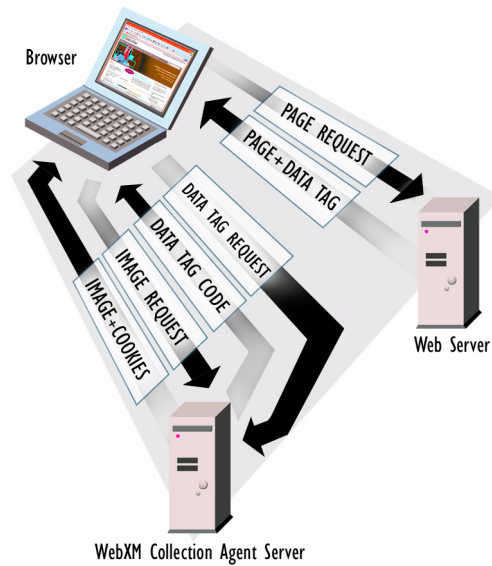


Figure 4-1 Collecting Visitor Information Using Data Tags (Steps 1 to 7)

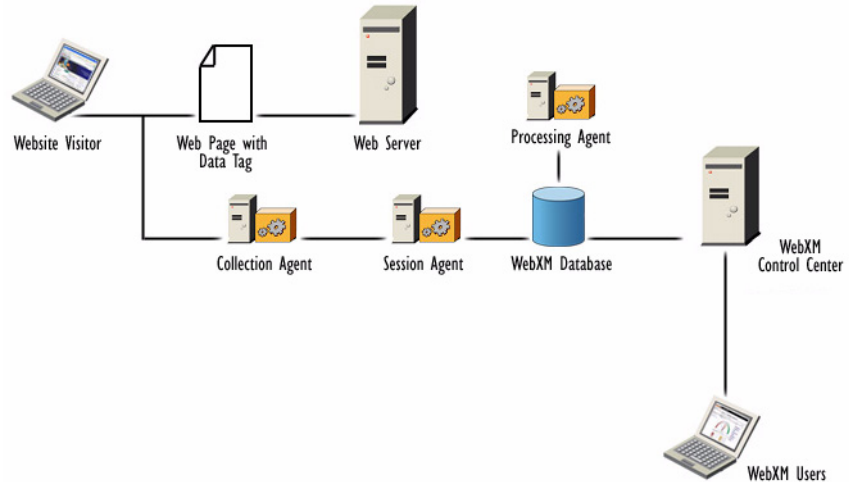


Figure 4-2 Assembling and Processing Visitor Information (Steps 8 to 11)

Scaling an Analytics Deployment

The Analytics module can be scaled to meet the needs of your organization. It supports multiple collection and processing agents as well as multiple instances of WebXM. Note that the session agent is not scalable — there can only be one per installation. For more details about the use of multiple instances of WebXM, see “WebXM on Multiple Servers” on page 2-14 and “Installing WebXM on Multiple Servers” on page 3-10.

Your decision whether to install one or a number of collection and processing agents depends on a number of factors, such as the number of web servers being tracked and the amount of traffic that you need to track on each web server.

When you install multiple agents, they can each be placed on their own server or they can all be placed on the WebXM server. Collection agents that have their own servers can take advantage of load balancing that may already be in effect for the website.

Multiple Instances of WebXM

The collection agent and the processing agent can support multiple instances of WebXM on the server where they are installed. For example, assume that an organization installs two instances of WebXM on a server. On the same server they install a collection agent, a session agent, and a processing agent. These agents can retrieve data from both instances of WebXM. The data from each instance is stored in its own database.

Guidelines for Load Balancing

As a general guideline, one collection agent can manage 150 million page views per month. This guideline varies, sometimes substantially, depending on several factors:

- The processing power of the server on which the collection agent is running. The estimate of 150 millions page views is based on the use of a server-class system with one Pentium 4 CPU running at 2 GHz. Other technical details, such as disk performance and amount of memory available can also affect the estimate
- The variability of the traffic to the web site; for example, you may have peak periods during which there are considerably more page views received than other periods

During the installation planning process our sales engineers will consider all of these factors and recommend a configuration that is appropriate for your environment.

Data Tag Syntax and Examples

A data tag consists of an HTML form and a JavaScript reference to a web application on the WebXM server. Together, the form and JavaScript represent only a few lines of code, as shown in the following example data tag:

```
<!-- Watchfire Analytics code start -->
<form name ="wf_identityForm">
  <input type="hidden" name="na"
  value="path=hierarchy_and_ID&dn=display_name&
  downloadTime=1">
</form>
<script language="JavaScript">
```

```

<!--
var wf_path = "http://collectionagentserver/instance_dc/";
//-->
</script>
<script language="JavaScript" src="http://collectionagentserver/
instance_dc"></script>
<!-- Watchfire Analytics code end -->

```

The HTML portion of the data tag can have one or two forms as follows:

- **wf_identityForm**, which can be used on all website pages where visitor data is collected. For more details about this form, see “Form Syntax for all Enabled Pages” on page 4-7
- **wf_searchresults**, which is used on the search engine results page to collect data about internal searches. For more details about this form, see “Form Syntax for the Search Results Page” on page 4-14



It is possible to omit the wf_identityForm portion of a data tag from a page. WebXM always automatically generates the data tag parameters that it needs to uniquely identify the page within reports when the form is not present. Although there are no settings required for WebXM to automatically generate a data tag, the page must still contain the JavaScript references. For more details, see “Using Automatic Data tags” on page 4-17.

If you plan to use the automatically generated data tags and your site uses query string parameters to identify pages, you may wish to configure the analytics job to also use the query strings. The job option **Use query string when generating path and display names in reports** is used for this purpose. For more details about this analytics job option, refer to the *WebXM User Guide*.

Form Syntax for all Enabled Pages

The wf_identityForm code can be used on all pages where visitor data is collected. If it is not present, then the path and display name parameters are automatically created; for more details see “Using Automatic Data tags” on page 4-17.

```
<form name ="wf_identityForm">
<input type="hidden" name="na"
value="path=page_path_and_uniqueID&dn=display_name&
downloadTime=1">
</form>
```

Each time you place wf_identityForm on a web page, some of its parameters must change, as indicated by the bolded items in the sample. They must change to identify the page within WebXM's reports and to facilitate the collection of visitor data. The following describes the form's parameters:



path The path parameter identifies the page for visitor calculations and for reporting purposes. The path parameter has two portions to it — a content category and a page ID — which look like the following inside a data tag.

Content Category

Page ID

Every page enabled with a data tag is uniquely identified by its domain and the data tag's path parameter. The domain is that of the page's URL,

unless the load balancing server option is in effect. If the load balancing server option is in effect, the domain is the parent domain; for example, a URL of *www.knotworks1.com/products/oak.htm* would be considered to have a domain of *www.knotworks.com*.

The path parameter usually corresponds to the folder in which the page resides. For example, a website's home page and two of its product pages might have the following paths:

- Home page path equal to *Knotworks*
- First product page path equal to *Knotworks/products/oak.htm*
- Second product page path equal to *Knotworks/products/pine.htm*

In the last two examples, the content category would be *Knotworks/products* and the page IDs would be *oak.htm* and *pine.htm*, respectively.








Sometimes pages are not stored in folders, as is the case for some content management systems. In this event, the path can be generated by inheriting the page structure of the content management system.

dn The optional *dn* parameter identifies the name of the page. The page name is used in WebXM's reports to represent a page or group of pages to Report Consumers as shown in Figure 4-3. It is therefore important to plan the *dn* parameters so that Report Consumers can identify a page or area of the website if they need to perform an action on it.

If the *dn* is not specified, the last portion of the path parameter is used in its place to identify the page. For example, in Figure 4-3 the path is *watchfire-sql/partners*, but if the *dn* is missing for the page then 'partners' would be used as the *dn* in the Subcategories and Pages area.

The string `watchfire-sql/partners` is the path parameter for the current page.

Details			
Category	Seconds	About	View
 All Domains > watchfir-sql > partners	0.50	-	-

Subcategories and Pages			
	Seconds	About	View
 Partner Program	1.00		 

Showing results 1-1 of 1 (Page 1) ◀ First | ◀ Previous | Next ▶ | Last ▶

Show results per page.

To produce this entry in the report, the page's data tag has a dn parameter of 'Partner Program'.

Figure 4-3 Average Time at Page Report Showing How Path and Dn Parameters Would Appear

downloadTime The optional `downloadTime` parameter is used when you wish to enable a page for download time measurement. A value of 1 indicates that download time measurement is on. A value of 0, or the absence of this parameter, indicates it is off.

Download time measurement requires the use of an **onload** event in the page. If your page does not use the onload event, then the data tags will automatically add such an event. If your page already uses the onload event for another function, the data tags will need to integrate with it. If you do not integrate with the existing event, it will be overwritten by the data tag's onload event. Contact Watchfire Professional Services to perform this integration.

Form Syntax for Marketing Campaigns

A marketing campaign is an effort made off-site to draw visitors to a website. A campaign landing page is a page that visitors arrive at only because they have entered through a marketing campaign. An example might be a magazine or radio advertisement that directs users to *http://site.com/radio*.

Marketing campaigns are tracked by WebXM using one of the following methods:

- 1** By a referring URL.
- 2** By a visit to a particular campaign landing page. This method is best used when the page will always belong to a campaign. When the page is removed, the campaign no longer exists (although its historical data remains).
- 3** Through a query string added to a visitor's URL. This is the optimal method to use in most cases because it does not need to come from or land on a specific page, nor do you need to change a site's data tags to implement it.

The method you choose depends on how you prefer to set up campaigns on your site and on the type of marketing campaign. For instance, an email campaign could either be tracked using a query string or by visitors arriving at a campaign landing page. An email campaign could not be tracked by a referring URL.

Before you begin implementing a campaign it is recommended that you first choose a method based on your needs and then plan how your campaigns will be portrayed within WebXM's reports. Careful planning up front helps ensure that you do not need to make additional changes to your website.

Bear in mind that the reports do not explicitly identify campaigns by their method. It is up to the Job and Report Administrator to determine the language that will be used in the report to identify a campaign. The Job and Report Administrator would do this as part of configuring an analytics job.

Restrictions for all Campaign Methods

Regardless of the method chosen, there are a few restrictions to the implementation of campaigns:

- The campaign landing page must have a data tag inserted within it; however its parameters will change depending on the method chosen
- The campaign landing page must be the first page with a data tag in the browser session, otherwise it will not be recognized as a campaign



Use a single method for your campaign. If you use more than one, a single user session would be entered into multiple campaigns.

A Campaign With a Referring URL

A campaign with a referring URL is enabled and configured through an analytics job. It does not have a query string parameter, nor does it require that the data tags on your website be modified. Refer to the *WebXM User Guide* for a description of how to track a campaign with a referring URL.

A Campaign With a Visit to a Landing Page

A campaign can be tracked solely through a particular landing page. When a visitor arrives at the page it has a data tag that identifies it as a campaign landing page. This method of tracking campaigns is best used when the page will always belong to a particular campaign. When the page is removed, the campaign no longer exists (although its historical data remains).



The first page of the session must be the campaign landing page.

Data Tag Parameters for a Landing Page

The data tags on campaign landing pages have an additional parameter added to them. The campaign ID, which is identified by the ‘wc’ parameter, is shown and described here:

```
<!-- Watchfire Analytics code start -->
<form name = "wf_identityForm">
<input type=hidden name="na" value="path=Home Page/Products/
Pine/Mouldings.htm&wc=PineMouldingCampaign">
</form>
```

A Campaign Having a Query String

A campaign that is tracked through a query string occurs when a visitor arrives at a page whose URL contains a query string parameter. This method of tracking campaigns has the following requirements:

- A standard data tag must be inserted into the page, as described in “Form Syntax for all Enabled Pages” on page 4-7. Any enabled page can therefore become a campaign landing page
- The page must be the first one of the session
- The link to the campaign landing page must have a query string that takes the following form: ?wc=MyCampaign, where **wc** is the query string parameter and **MyCampaign** is the campaign ID
- The campaign must be configured as part of the analytics job in order to collect and report on it. The configuration includes defining the campaign ID at the *[Analytics Job Name] > Report Data Collection > Campaigns* page. For more details about configuring an analytics job, refer to the *WebXM User Guide*

Example: Multiple Campaigns Using Query Strings

The query string method is useful if you have multiple email campaigns. Each link within an email could have a different query string value. The following example shows three different email campaigns, each with their own query string:

www.knotworks.com/products/pine/mouldings.htm?wc=mouldingcampgn

www.knotworks.com/products/oak/bannisters.htm?wc=bannistercampgn

www.knotworks.com/products/mdf/baseboards.htm?wc=basemdfcampgn

Each of these links represents a separate email campaign that is tracked by a single analytics job. When configured as part of a job, each campaign will show separate results in the reports. The advantage to this method is that

you can add campaigns to your site whenever you wish. All you need to do is create a new query string and configure the job accordingly.

To configure an analytics job for Knotworks' campaigns using a query string:

- 1 Go to the Webspace Center and click **Create** an analytics job.
- 2 Configure the campaign at *Analytics Job > Report Data Collection > Campaigns*. You can create multiple campaigns and use the job to set up how you wish them to be represented within the reports.

Form Syntax for the Search Results Page

When you wish to track the internal search engine results, an additional form is added to the data tag on the search results page. The form tracks the keywords leading to the search results. The data tag should only appear on the first page of any multi-page search results to avoid duplicate submissions. The format of the data tag for the search engine results page should be as follows:

```
<!-- Watchfire Analytics code start -->
<form name="wf_searchresults">
<input type="hidden" name="iseterms" value="searchTermList">
<input type="hidden" name="isehits" value="numberOfHits">
</form>

<form name="wf_identityForm">
<input type="hidden" name="na" value="path=Search/
SearchResults">
</form>

<script language="JavaScript">
<!--
var wf_path = "http://collectionagentserver/instance_dc/";
//-->
</script>

<script language="JavaScript" src="http://collectionagentserver/
instance_dc/"></script>
<!-- Watchfire Analytics code end -->
```

Data Tag Parameters for Search Results

Before implementing the search results data tag, refer to the following descriptions of its search term list and hit counter parameters.

searchTerm List	The value searchTermList refers to a list of all search terms, separated by exclamation points, with a terminating exclamation point. This parameter is populated by the execution of the data tag on the search results page. Note that if the exclamation mark character (!) is part of a search term, it must be encoded as %21. When you are creating a search results form, ensure that this encoding is handled properly.
numberOfHits	The value numberOfHits refers to the number of matches returned by the search engine. This parameter is populated by the execution of the data tag on the search results page.

In cases where it is not possible to know the total number of matching search results, it is acceptable to return 1 for a successful search and 0 for an unsuccessful search that returned no hits.

The following example shows two search terms, product and specifications, that generated 435 hits:

```
<form name="wf_searchresults">
<input type="hidden" name="iseterms"
value="product!specifications!">
<input type="hidden" name="isehits" value=435>
</form>
```

JavaScript Syntax for all Pages

A data tag always includes JavaScript whose purpose is twofold:

- Reference the location of the web application, known as the collection agent. Typically, the collection agent is located on the WebXM server, but it could be installed on its own server
- Reference the script location, whose purpose it is to collect visitor information from the site and send it to the collection agent. The script could be located on the WebXM server or on its own server

With the use of JavaScript comes the restriction that 'JavaScript' must always be turned on in a visitor's browser before data can be collected from that visitor.

The JavaScript reference to the collection agent on the WebXM server is as follows:

```
<script language="JavaScript">
<!--
var wf_path = "http://collectionagentserver/instance_dc/";
//-->
</script>
```

The JavaScript reference to the script location is as follows:

```
<script language="JavaScript" src="http://collectionagentserver/instance_dc"></script>
```

If you are implementing data tags on https pages, then you would use the following JavaScript references:

```
<script language="JavaScript">
<!--
var wf_path = "https://collectionagentserver/instance_dc/";
//-->
</script>
<script language="JavaScript" src="https://collectionagentserver/instance_dc"></script>
```

The values to use for the collection agent server will be supplied to you during installation either by Watchfire support or sales engineering staff.

Special Characters in a Data Tag

Data tag attribute values can use any ASCII character except the double quote and ampersand. These need to be encoded as follows: the double quote as %22 and the ampersand as %26.

In the search results form, the search keywords can use any ASCII character except the exclamation mark, which needs to be encoded as %2e, the double quote, and the ampersand.

Example: Encoded Character in wf_identityForm

Here is an example that uses the encoded ampersand character:

```
<form name="wf_identityForm">
<input type=hidden name="na" value="path=/dogs/food/
kibbles%26bits.htm&dn=kibbles%26bits.htm">
</form>
```


Methods for Populating a Website With Data Tags

The methods used to populate your website content with data tags varies from one installation to another, depending on the platforms and technologies in use at your site.

To simplify their implementation and maintenance, data tags are typically generated and added to website content through one of the following methods:

- As server-side ASP/JSP code
- Through a content management system
- Through a centralized script that is run on a directory of files
- Manually

Best practices state that the data tags should be placed at the top of each page. Ensure that the JavaScript reference in the tag follows the HTML form portion.



For some content management systems, Watchfire has template code that can be used as is or modified to suit your requirement.

Using Automatic Data tags

Although you can implement the data tags as documented, the wf_identityForm portion of the data tag is automatically created by WebXM if it is not present. This means that WebXM creates both the path and display names for you by normalizing the URL of a page. Typically, you would only use the wf_identityForm portion if you wished to override the automatic creation. For example, if your site used a content management system that built its page URLs using cryptic IDs, then the normalized URL would not be suitable for automatically creating the path and display name parameters in your reports. In this case, you would implement the full data tag with the wf_identityForm portion.

Changing How Content is Automatically Generated

Every job contains an option that defines how the automatic creation of data tag content is performed. It is referred to as the **Use query string when generating path and display names in reports** option and

appears on the What to Track page. Whether or not you use the option depends on whether or not your website is based on directory instead of query string parameters. If your website is based on a directory, you should ignore the query string parameters and deselect this option.

When the option is selected, WebXM automatically generates path and display names in reports using the following methodology:

- The URL is normalized using the Session ID and case sensitivity settings of the job
- The `http://server:port/` portion of the URL is removed
- If present, the `?` character is replaced with a `/`
- The `//` characters are replaced with a single `/`

The path is the remaining string, while the display name is the last portion of the string up to the first `/`. Pages will appear in your reports according to their query string parameter, if present. For examples on how the query string option functions, see “Examples for Automatic Data Tags” on page 4-19.

When the option is deselected, WebXM automatically generates path and display names in reports using the following methodology:

- The URL is normalized using the Session ID and case sensitivity settings of the job
- The `http://server:port/` portion of the URL is removed

The path is the remaining string, up to the `?` character, if present. In other words, there are no query strings appearing in your reports. If your website is based on directory instead of query string parameters, you should deselect the option so as to ignore the query string parameters.

Examples for Automatic Data Tags

In Table 4-1, the URL is that of a sample page, the path defines how Report Consumers will navigate to the page within reports, and the display name is the name assigned to the page within reports, enabled indicates that the **Use query string** option is selected, and disabled indicates that it is turned off.

URL	Use Query String Option	Path	Display Name
http:// www.watchfire.com/ Products/WebXM/ gottagetit.asp?sid=1234&a=1	Enabled	Products/WebXM/ gottagetit.asp/a=1	a=1
	Disabled	Products/WebXM/ gottagetit.asp	gottagetit.asp
http:// www.watchfire.com/ products/index.htm	Enabled	Products/index.htm	index.htm
	Disabled	Products/index.htm	index.htm
http:// www.watchfire.com/ Products/	Enabled	Products/(not specified)	Products/(not specified)
	Disabled	(not specified)	(not specified)

Table 4-1 Enabling and Disabling the ‘Use Query String’ Option

Job Properties Affected by Data Tags

Certain analytics job properties require you to know the contents of a particular data tag. They are as follows:

- The What to Track page’s **Use query string when generating path and display names in reports** option
- The Outcomes page, when adding outcomes to it
- The Tracked Paths page, when adding paths to it

To easily find the contents of the data tag for a given page, use the Analytics Data Tag report. This report is available through the WebXM Explorer Bar.

If You Have a Defined Path Structure

For sites whose pages have a defined path structure in the URL, the wf_identityForm HTML is typically generated to match the path structure. For example, a page at */company/pressreleases/2004/index.html* would have the following:

- A path of */company/pressreleases/2004/*
- A dn of *index.html*

If You Have no Path Structure

Some sites have no path structure in the URL. For example, some content management systems generate URLs similar to the following: *http://site.com/content.asp?id=23989829438592345*.

In this case the solution is to integrate with the content management system's scripting abilities to generate the data tag automatically on each page and to populate its *path* and *dn* with values from the content management system itself. Watchfire has experience with a variety of content management systems and can supply templates and examples.

Deploying Analytics on Your Website: a Summary

Having read and understood how the Analytics module uses data tags to collect visitor information from your website, here is a summary of how to plan and deploy the module:

- 1 Determine what will be tracked for web analytics. Will it be a single website, all of your external web properties, or just a portion of them? The number of websites that you track will determine how the data tags are deployed. For example, if you are only interested in visitor data from your external website, and not your intranet, then you may deploy the data tags using asp or server-side code. In this situation, adding the data tags to a content management system would be unnecessary.
- 2 Determine the kind of visitor information you wish to collect. Do you wish to know about visitor behavior, such as visits, clickpaths, and exits, or do you wish to track your marketing campaigns to measure their Return-on-Investment? In some cases, the kind of information that you wish to track determines the configuration of the data tag.

For more details about the data tags available, see “Data Tag Syntax and Examples” on page 4-6.

- 3 Determine how your analytics data will be represented within WebXM’s reports. The naming convention you apply to the path and dn parameters of a data tag affects how the data is presented within reports. For examples on how to use these parameters, see “Data Tag Syntax and Examples” on page 4-6 and the *WebXM User Guide*.
- 4 Add the data tags to your development or staging servers. The data tags should be deployed outside of your production environment so that they can be tested first. For more details about inserting data tags into your website content, see “Methods for Populating a Website With Data Tags” on page 4-17.
- 5 Test the implementation to ensure that it is producing valid data:
 - Configure the analytics job to collect and report on the data you require. An analytics job must be created and configured before you can generate reports. For more details about the analytics job, refer to the *WebXM User Guide*
 - Set up and generate reports containing an analysis of the web analytics data collected
- 6 Add the data tags to your production environment and repeat step 5.

THE WEBXM DATABASE

Topics

- Overview
- Estimating the Ultimate Size of the WebXM Database
- Using a SQL Server Database with WebXM
- Using an Oracle Database with WebXM

Overview

This chapter provides information about the deployment and configuration of the WebXM database in environments where SQL Server 2000 or Oracle are being used. It will be of particular interest to application developers, consultants, WebXM system administrators and database administrators.

The WebXM Server stores the information gathered during a scan in the database. You can use a Microsoft SQL Server 2000 database or an Oracle database; the one you choose depends on how you plan to implement WebXM.

Estimating the Ultimate Size of the WebXM Database

The size of the WebXM database is primarily a function of these factors:

- The size and composition of the websites being scanned
- The number of links on pages
- The number of issues found on the websites
- The number of distinct content scan jobs defined

Secondarily, the size of the WebXM database is a function of these factors:

- The number of users
- The number of reports that are defined

To estimate the size of a WebXM database, allow 10 kB of storage for each URL to be scanned, assuming an average of 20 links per page. If your pages have more than 20 links each, then multiply the 10 kB of storage per link by the multiple of 20 links per page. For example, if your pages have 100 links per page, then you should allow 50 kB of storage for each URL to be scanned.



The number of URLs includes all HTML pages and, in addition, all page content such as images, documents and multimedia files and all links to other resource such as mailto and ftp links.

This estimate can be refined by running a test content scan job of 10,000 or more URLs and calculating the amount of storage used per URL.



If you are planning to collect code fragments during an accessibility scan, we recommend the following guidelines when estimating the space required for the database:

- If you are collecting accessibility data during a scan (no code fragments), estimate 10 - 15 kB of space per URL being scanned
- If you are collecting accessibility data with code fragments, estimate 15 - 20 kB of space per URL being scanned

Database Security

WebXM is architected such that only a single service account is required to have access to the database. Individual WebXM users do not require any form of database permissions.

WebXM users are authenticated on the basis of their Windows log-on credentials. Individual user rights (to webspaces) and role definitions are stored in database tables and are keyed to the user's Windows domain account name.

For more information about user accounts and roles, refer to the *Administration Guide*.

Using a SQL Server Database with WebXM

The WebXM database contains all administration, configuration and reporting data. In SQL Server, a distinct database is created during the WebXM configuration. This database contains all the table definitions, indexes, constraints, and database stored procedures used by WebXM.

WebXM uses database tables in four ways:

- 1 To store data that is independent of a particular content scan job.

These types of tables are named in mixed case with the name denoting the data that is stored in the table; for example: Job, UserInfo.

- 2 As a template for tables that are used to create tables that store data for a particular content scan job.

These tables have the suffix **_JII_** appended to their names; for example: UrlInfo_JII_, UrlReference_JII_.

- 3 To store data for a particular content scan job iteration.

These tables are created (based on the corresponding template table's definition) the first time that the job is run. The table names consist of the template table name with the **_JII_** suffix replaced by the job identifier and job iteration; for example: UrlInfo_32_1, UrlReference_32_1.

- 4 To store default options for a particular content scan job.

This is a single table, per content scan job, that is created the first time the content scan job is created. The table name is **JobOption**, followed by the job identifier and the string **_D**; for example: JobOption_310_D.

Similarly, there are three types of stored procedures in WebXM:

- 1 Stored procedures that perform operations that are independent of any particular content scan job.

The names of these stored procedures begin with the prefix **wp_** and are named according to the operation that they perform; for example: wp_InsertCount, wp_Job_Delete.

- 2 Template stored procedures that are used to create stored procedures that perform an operation on the data for a particular content scan job.

The names of these stored procedures begin with the prefix **wt_**; for example: **wt_Mime_Insert**, **wt_UrlInfo_Insert_New**.

- 3 Stored procedures that perform operations on the data for a particular content scan job iteration.

These stored procedures are created from the template stored procedures the first time a content scan job is run. The name of the stored procedure begins with the prefix **wi_**, followed by the job identifier, job iteration, and operation name; for example: **wi_21_0_Mime_Insert**, **wi_21_0_UrlInfo_Insert_New**.

Referential integrity in WebXM is performed at the database level. All foreign key constraints are defined in the database.

SQL Server Configuration

There are a number of SQL Server 2000 configuration options that can affect the performance of WebXM. These options are defined using SQL Server Enterprise Manager.

To define SQL Server Properties:

⇒ Right-click on the server name.

To define Database Properties:

- 1 Select the server name.
- 2 Expand **Databases**, and then right-click on the database name.

Memory

These options, available on the Memory tab of the SQL Server Properties dialog box, control the amount of memory that SQL Server will use. You must be very careful about allocating too little or too much memory to SQL Server. Too little memory will negatively impact SQL Server performance. Too much memory may result in SQL Server taking essential resources away from other applications, such as the operating system, causing a reduction in overall system performance.

The recommended settings for these options are:

Dynamically configure SQL Server memory: Select this option. Using dynamic memory allocation, SQL Server can add memory to handle incoming queries, free up memory for another application you're starting, or reserve memory for possible needs.

Reserve physical memory for SQL Server: Do not select this option; it can have a negative performance impact on other applications running on the server.

Minimum query memory: The default value of 1024 kB is sufficient.

Security

These options, available on the Security tab of the SQL Server Properties dialog box, control user authentication and auditing.

The recommended settings for these options are:

Authentication: Select the **Windows only** option. Only users with a domain account can access the server.

Processor

These options, available on the Processor tab of the SQL Server Properties dialog box, control how and when processes are used. The default settings are the recommended settings.

SQL Server will use multiple processors on behalf of WebXM when running multiple simultaneous content scan jobs and when multiple users are viewing reports simultaneous.

Server Settings

These options, available on the Server Settings tab of the SQL Server Properties dialog box, control the general behavior of the database server.

The recommended settings for these options are:

Default language for user: This option must be set to **English**.

Allow triggers to be fired which fire other triggers (nested triggers): Select this option. Nested triggers are useful for executing a series of tasks within a single transaction. For example, an action can initiate a trigger that starts another trigger, which in turn can start another trigger, and so on. Because the trigger is handled within a transaction, a failure at any level causes the entire transaction to roll back, which reverses all changes

to the database. As a fail-safe measure, triggers are terminated when the maximum nesting level is exceeded. This protects against an infinite loop.

Use query governor to prevent queries exceeding specified cost: Do not select this option. The Query Governor prevents you from running any queries that have a running time that exceeds a specified query cost. By default, the Query Governor is turned off, meaning there is no maximum cost. Enabling this option will cause WebXM queries to be cancelled since they tend to take up more resources.



The default values can be used for all other options in the SQL Server Properties dialog box.

Database Properties

These options, available on the Options tab of the Database Properties dialog box, define the characteristics of the selected database.

The recommended settings for these options are:

Recovery Model: Set this option to **Simple**. By default, this is set during configuration of WebXM. The database can be recovered only to the last full database backup or last differential backup.

Auto create statistics: Select this option. This option allows queries to be optimized, thus improving performance.

Auto update statistics: Select this option. Statistics can become out-of-date when the data in the tables has changed. This options allows existing statistics to be updated, allowing queries to be optimized.

Full-text Search Indexing

Full-text search indexing allows you to perform language text searches on the SQL Server database. However, this feature can affect the overall performance of the database. WebXM does not use full-text search indexing capabilities; this service should be turned off.

Database and SQL Log File Configuration

You can improve WebXM performance by considering the following items when configuring the database and log files:

- 1 Pre-allocating the space required for database files and log files will improve performance. These options are available on the Data Files tab and the Transaction Log tab of the Database Properties dialog box in SQL Server Enterprise Manager.
- 2 Allowing the log files to grow automatically (by 10%) is required to ensure that unexpected errors do not occur.
- 3 Placing the data files and log files on separate physical disk drives will improve performance substantially. Make sure that these physical disk drives have enough free space to allow for database growth.

SQL Server Database Backup and Maintenance

Like any enterprise application, the WebXM database must be backed up regularly, and other database maintenance tasks must be performed from time to time. SQL Server Enterprise Manager provides two wizards that allow these tasks to be automated:

- Create Database Backup Wizard
- Database Maintenance Plan Wizard

It is recommended that these wizards be used to create the required scheduled tasks.

Backup Strategy

Because the database log files can grow in size between backups, it is recommended that the WebXM database be backed up daily. Depending on the frequency with which content scan jobs are run, it may be possible to do incremental backups frequently and full backups less frequently. It is not necessary to perform backups while the database is completely quiet, but backup operations should be scheduled for times when the database is known to be less busy. If your organization employs a regular maintenance window for servers, then this may be an ideal time to perform the backup.

For large organizations, where the database is never, or rarely, quiet, consider using commercial backup software, configured to backup SQL incrementally.

Database Recovery

In the event of a catastrophic hardware failure, the WebXM database can be restored from the last backup by using the 'Restore database' command in SQL Server Enterprise Manager.

Database Maintenance

Once WebXM has been successfully installed, a database maintenance plan must be established. Use the 'Database Maintenance Plan Wizard' to create the plan and schedule it. This wizard can be used to:

- Ensure that the database optimization statistics are up to date
- Verify database integrity
- Configure a database backup
- Generate administrative reports

Shrinking the Database

Database growth can become an issue, especially after large content scan jobs have been deleted. The 'Shrink Database' command can be used to remove the empty space. The database is most effectively shrunk at the "File" level. Choose "Files" from the "Shrink Database" window.

Alternatively, the Database Maintenance wizard can be configured to periodically shrink a database.

Disk Defragmentation

Disk fragmentation occurs over time as files are created, deleted, and change in size. Consider using the Windows tools to periodically defragment disks when the database is not being used and can be taken down for maintenance.

Monitoring Database Performance

There are a number of tools available that allow you monitor the performance of a scan.

Windows Performance Monitor (Perfmon)

WebXM provides some Performance Monitor counters that can help you track your scan progress.

To add these counters to Perfmon:

- 1 Selecting the WebXM Content Agent performance object in Perfmon.
- 2 Select the counters and then select the instance you want to monitor. The instance has the same name as the scan job that is executing. The counters are described in the *WebXM User Guide*.

Processes to Monitor

Using the Process performance object, you can select the WFAgentHost and SQLserver instance, and then choose the % Processor Time counter. This will tell you how much of the system's CPU each of these processes are using. If SQLServer or the WFAgentHost are constantly above 90%, you should consider adding another CPU to your system or increasing your CPU speed.

Troubleshooting SQL Server Database Issues

A number of tools are available to assist with diagnosing and troubleshooting database issues. This section describes these tools and some circumstances under which it may be necessary to use them.

SQL Server, Windows and WebXM Tools

The main tools that are available to diagnose and troubleshoot database issues are:

- SQL Server Enterprise Manager — the SQL Server administration tool
- Query Analyzer — a SQL Server tool that allows SQL queries to be entered and executed
- Windows Event Log — SQL Server writes error information to the event log under some circumstances
- Windows Performance monitoring tool — which provides the ability to monitor WebXM and SQL Server performance statistics

Common Problems and their Symptoms

If jobs are not completing, or are being placed in a Suspended state, start by checking the entries in their log files. To access a job's log file, find the job's thumbnail in the webspace center and click the **Log** link.

Also check for errors in the Windows Application Log using Windows Event Viewer.

Make sure that there is sufficient free disk space on the SQL Server machine.

Process Blocking

If a scan job is running, but the number of Checked Links does not seem to increase over several minutes, there may be some process blocking on SQL Server.

To locate blocking processes:

- 1 Using SQL Server Enterprise Manager, expand the Management folder under your server.
- 2 Expand the Current Activity section.
- 3 Expand the Locks / Process ID section.

If there are any blocking processes, they will show up as red icons and have the word (Blocking) next to the spid number.

It is normal to have blocking processes for short periods of time; however, if your server is blocking for extended periods, please contact Customer Support for further help with this issue.

Using an Oracle Database with WebXM

The WebXM database contains all administration, configuration and reporting data used by WebXM. In Oracle, a separate UserId should be created to own all the WebXM objects. This UserID contains all the table definitions, indexes, constraints, and all database packages used by WebXM.

WebXM uses database tables in four ways:

- 1 To store data that is independent of a particular content scan job.

The names for these types of tables denote the data that is stored in the table; for example: Job, UserInfo.

- 2 As a template for tables that are used to create tables that store data for a particular content scan job.

These tables have the suffix **_JII_** appended their name; for example: UrlInfo_JII_, UrlReference_JII_.

- 3 To store data for a particular content scan job iteration.

These tables are created (based on the corresponding template table's definition) the first time that the job is run. Their name consists of the template table name with the **_JII_** suffix replaced by the job identifier and job iteration; for example: `UrlInfo_32_1`, `UrlReference_32_1`.

- 4 To store default options for a particular content scan job.

This is a single table, per content scan job, that is created the first time that the content scan job is created. The table name is **JobOption**, followed by the job identifier and the string **_D**; for example: `JobOption_310_D`.

Similarly, there are three types of packages in WebXM:

- 1 Packages with stored procedures that perform operations that are independent of any particular content scan job.

The names of the packages are prefixed with **wpkg_**. The names of these stored procedures within the package begin with the prefix **wp_** and are named according to the operation that they perform; for example: `wpkg_AdminJob.wp_Job_Delete`.

- 2 A template package that is used to create packages that perform an operation on the data for a particular content scan job.

The name of this package is `wtkg_AgentContent`.

- 3 Packages that perform operations on the data for a particular content scan job iteration.

These packages are created from the template package the first time a content scan job is run. The package names begin with the prefix **wikg_**, followed by the job identifier, job iteration, and operation name; for example: `wikg_21_0_AgentContent`.

Referential integrity in WebXM is performed at the database level. All foreign key constraints are defined in the database.

Oracle Transaction Type

WebXM uses the following Oracle transaction types:

- OLTP
- Reporting
- Data Warehouse / DSS / OLAP

Oracle Database Server Installation

If you are planning to store scan data in an Oracle database, there are a number of tasks you must perform in Oracle before you can configure WebXM.

Step 1: Install the Oracle Client on each WebXM Server machine; for example, on each Agent server and WebXM Control Center. This tool provides the connection between the Oracle database server and the WebXM server.

Step 2: Install the Oracle Provider for OLE DB Release 9.2.0.4 or OLE DB Release 10.1.0.2 on each WebXM Server machine; for example, on each Agent server and WebXM Control Center.

The host string, usually an alias for the Oracle server, requires the username and password to connect. It can be set up in the *tnsnames.ora* file, found in `\oracle\ora92\network\ADMIN` (for Oracle 9i) or in `\oracle\product\10.1.0\Client_1\network\ADMIN` (for Oracle 10g) on the machine connecting to Oracle. As long as the WebXM server can access the IP associated with the host string, it should be able to connect to the Oracle database server.

Step 3: On the database server, using the Oracle Database Configuration Assistant, create a new database.



When you create a new database, choose the **Locally managed tablespaces** option, rather than **Dictionary managed tablespaces**. This will improve the performance of the database. For more information, see “Oracle Database Configuration” on page 5-15.

Step 4: Using the Oracle Net Configuration Assistant, create a Listener. You can use the existing Listener that was configured for an existing database.

Step 5: Using the Oracle Net Configuration Assistant, create a new Net Service on each client machine.

Step 6: On the database server, using the Oracle DBA Studio, create a new tablespace in the database, where the WebXM data will be stored.

The size of the tablespace depends on the size of your website and the number of links you will be scanning with WebXM. For more information, see “Estimating the Ultimate Size of the WebXM Database” on page 5-2.

Step 7: On the database server, using the Oracle DBA Studio or SQL Plus, create a user and explicitly grant the following permissions to the user:

- Alter Any Table (for upgrade only)
- Create Table
- Create Procedure
- Create Sequence

Associate this user with the new tablespace you created.

Ensure this user also has the following standard roles: Connect and Resource.



You cannot achieve the permissions through a “Role” in Oracle. WebXM employs PL/SQL (a procedural language extension to Oracle), which requires explicit permissions since it does not look for role membership.

If the user does not have the correct privileges, the following error will be recorded in the log file.

```
ORA-01031: insufficient privileges
```

Also, the user will not be able to view and create jobs or reports.

Client Requirements

All client machines that will be running WebXM will require Oracle's OLE DB Provider version 9.2.0.4 or OLE DB Provider version 10.1.0.2. In addition, the Oracle Client must be installed on each WebXM Server machine; for example, on each Agent server and WebXM Control Center. This tool provides the connection between the Oracle database server and the WebXM server.

Using the Oracle Net Configuration Assistant, create a new Net Service to point to the database already created on the database server.

Oracle Database Configuration

You can improve WebXM performance by considering the following items when configuring the database:

- ⇒ When creating the database, enable the **Locally managed tablespaces** option. As the database grows, storing more data with each scan, the amount of space available for the database is extended. An initial table space allocation of 32 kB is recommended.



Space available for the database is only extended during normal writing operations; not during database creation. Therefore, if the initial allocation for the database is too small, you will not be able to run the Configuration Wizard to create the WebXM database.

Controlling Dynamic Tables in Oracle Databases

Oracle database administrators may want more control over where tables and indexes are created for WebXM. It is often standard with Oracle databases to have tables in a tablespace on one drive and indexes in a separate tablespace on another drive. WebXM provides you with this kind of control over dynamic tables.

In WebXM, `_JII_` tables or indexes are templates used by a content scan job when it creates tables or indexes in the database. If you move either template to a different tablespace, the tables or indexes will get created on that tablespace whenever the content scan job is itself created.



This procedure should not be attempted by anyone other than an Oracle Database Administrator.

To implement dynamic tables, perform the following:

- 1 Install WebXM and run the Configuration Wizard.
- 2 Move the `_JII_` tables or indexes to the tablespace that you want the job iteration tables to be created in (usually drop a re-create).
- 3 Restore any constraints that were affected by moving the tables.

- 4 Re-run the Configuration Wizard to re-validate stored procedures, as they would have become invalid when the tables were deleted.

Now, when you create a content scan job in WebXM, the new tables for the job will be created in the same tablespace as the _Jll_ template tables or indexes.

Characterization of Database Growth

WebXM will cause the Oracle database to grow 15 kB per URL scanned. Multiply this by two for each job that is run multiple times, since WebXM keeps both a working and a reporting set of data.

Oracle Database Backup and Maintenance

Like any enterprise application, the WebXM database must be backed up regularly and some database maintenance tasks must be performed from time to time

Backup Strategy

It is recommended that the WebXM database be backed up daily. Depending on the frequency with which content scan jobs are run it may be possible to do incremental backups frequently and full backups less frequently. It's not necessary to perform backups while the database is completely quiet but backup operations should be scheduled for times when the database is known to be least busy. If your organization employs a regular maintenance window for servers then this may be an ideal time to perform the backup.

For large organizations, where the database is never, or rarely, quiet, consider using commercial backup software, configured to backup Oracle incrementally.

Database Recovery

In the event of a catastrophic hardware failure, the WebXM database can be restored. Refer to the Oracle documentation for more information.

Database Maintenance

Once WebXM has been successfully installed a database maintenance plan must be established. Use the plan to ensure that the database optimization statistics are up to date, to verify database integrity, configure a database backup and generate administrative reports.

Disk Defragmentation

Disk fragmentation occurs over time as files are created, deleted, and change in size. Consider using the Windows tools to periodically defragment disks when the database is not being used and can be taken down for maintenance.

Moving an Oracle Database with WebXM

If a database needs to be moved from one Oracle install to another, it is recommended that you use the Export/Import utilities within Oracle. The Export utility will export the tablespace to a file, while the Import utility will import it to a new server.

Monitoring Database Performance

There are a number of tools available that allow you monitor the performance of a scan.

Windows Performance Monitor (Perfmon)

WebXM provides some Performance Monitor counters that can help you track your scan progress.

To add these counters to Perfmon:

- 1 Selecting the WebXM Content Agent performance object in Perfmon.
- 2 Select the **Links Found** and **Links Checked** counters, and then select the instance you want to monitor. The instance will have the same name as the scan job that is executing.

Processes to Monitor

Using the Process performance object, you can select the WFAgentHost and oracle instance, and then choose the % Processor Time counter. This will tell you how much of the system's CPU each of these processes are using. If Oracle or the WFAgentHost are constantly above 90%, you should consider adding another CPU to your system or increasing your CPU speed.

Troubleshooting Oracle Database Issues

A number of tools are available to assist with diagnosing and troubleshooting database issues. This section describes these tools and some circumstances under which it may be necessary to use them.

Oracle, Windows, and WebXM Tools

The main tools that are available to diagnose and troubleshoot database issues are:

- DBAStudio — the Oracle administration tool
- SQLPlus or SQLPlus Worksheet — an Oracle tool that allows SQL queries to be entered and executed
- Windows Event Log — Oracle writes error information to the event log under some circumstances
- Windows Performance monitoring tool — provides the ability to monitor WebXM and Oracle performance statistics

C H A P T E R

6

TIPS, TRICKS, AND TROUBLESHOOTING

Topics

- Overview
- Tips & Tricks
- Q & A
- Troubleshooting WebXM

Overview

This chapter provides a few tips to help you achieve the best performance from the WebXM Servers and the WebXM Desktop applications, and information to help you solve any problems you may encounter while installing and configuring WebXM.

Tips & Tricks

This section provides tips to help you optimize and manage the performance of WebXM.

Upgrading WebXM

Here are a couple of things to keep in mind when upgrading WebXM:

Custom Options.xml

If you have a custom Options.xml file, it will not be retained during the upgrade. You can retain your custom settings by removing the custom option.xml file before beginning the upgrade. When you are finished upgrading WebXM, import the custom changes from the “old” options.xml file to the “new” one.

WebXM Server and WebXM Desktop on a Single Machine

If you have installed WebXM Server and WebXM Desktop on the same machine, it is best to upgrade the WebXM Server first. By doing this, you will avoid error messages related to the WebXM services.

Performance Tuning

WebXM Server and WebXM Desktop

One factor that affects the performance of a scan relates directly to the performance and capacity of the hardware on which it operates. By optimizing the scanning PC, you can minimize the amount of time required to successfully complete the scans and generate the various reports that have been requested by stakeholders.

The following are issues and possible alternatives to addressing these issues. These alternatives will have the greatest impact on the

performance of both the time to perform a scan, post processing and report generation. It is important that you review this material with the appropriate technical resources within your own organization for their input. Watchfire's Professional Services team would be pleased to assist you in discussing and or implementing any of these alternatives or to answer any of your questions.

Proximity of the scanning PC to the web server

The greater the number of hops and latency along the internet backbone between the scanning computer and the web server, the greater the time (latency) required to perform the scan. You should consider the benefits of locating the scanning computer on the same internet backbone (in the same environment as the web server you are scanning). Try to put the scanning computer on the fastest internet access (least contention with other users) and as close (in hops) as possible to the web server it is scanning.

Size of the scan areas

If possible, configure WebXM to scan large websites in logical sections as opposed to one large scan.

Scanning hardware configuration

Having additional memory, processors and fast storage devices, as well as how the software is installed and the operating system configured, can optimize the performance of the scanning computer.

CPU Configuration

To maximize performance, you should ensure that the scanning computer is dedicated to the purpose of performing the scan, analysis, and report generation, from start to completion.

Report Generation

It is important to note how certain reports are generated and what impact they have on time to complete. For example, the Deep Pages report and the Broken Anchors report require a great deal of processing time as they must sequentially and iteratively pass through the database many times to generate the required information. You should therefore only request the specific reports that you feel are important in order to minimize the time required to generate these reports.

Executing JavaScript

JavaScript execution can adversely affect the performance of a content scan, thus tying up one of the available Agents. If you plan to execute any JavaScript encountered during a content scan, consider installing additional agent servers.

WebXM Database

For suggestions on tweaking the database for optimum performance, refer to the Microsoft SQL Server Performance Tuning guides or the Oracle documentation.

Verifying the Agent Service and Alerting Service Installation

During the installation of WebXM, two services are installed:

- Watchfire WebXM Agent Service
- Watchfire WebXM Alerting Service

You need to ensure that these services have been installed and are started. If the agent service is not started, any jobs that users create will not be picked up and run by WebXM Server. If the alerting service is not started, any alerts that have been configured for users will not be issued.

If you installed WebXM Server components on different machines, you must verify that the services are started on each one.

To verify that the services are installed and started:

- 1 Using the Control Panel or the Start Menu, select **Administrative Tools > Services**.
- 2 In the list of services, select **Watchfire WebXM Agent Service**. If the service was properly installed and started, **Started** will be displayed in the Status column.

If this is not the case, you can start the service by right-clicking on the service name and selecting **Start**.

- 3 Repeat Step 2 for the Watchfire WebXM Alerting Service.

Configuring Internet Explorer

If the web server uses chunked data, WebXM must force HTTP 1.0. The Internet Explorer setting must be changed for the user that performs the scans; for example, the Content login account and the WebXM service account. If you do not change the Internet Explorer settings, the scan engine cannot use HTTP 1.0, and will add the following error message to the job's log and then continue with the scan:

```
<domain-being-scanned> uses Transfer-encoding: chunked. Cannot force HTTP/1.0
```

If you are scanning a site that uses “chunked encoding” and you have the HTTP 1.1 setting selected in Internet Explorer, the scan will become very slow and possibly freeze. Once you disable the HTTP 1.1 setting, you must stop and then restart all currently running scans that appear to be using chunked encoding.

To force HTTP 1.0:

- 1 Log onto the WebXM server using the service account.
- 2 Open Internet Explorer.
- 3 Select **Tools > Internet Options**.
- 4 Select the Advanced tab.
- 5 Under HTTP 1.1 settings, clear the following check boxes:
 - Use HTTP 1.1
 - Use HTTP 1.1 through proxy connections

Configuring Internet Explorer for Advanced Login

If you will be recording interaction scripts using the Advanced Login feature, both the machine that records the script and the server used to run content scan jobs with Advanced Login enabled **must** use the same Internet Explorer options and settings of the AppScan Enterprise service account.



(Windows 2003 or higher) - Uninstall IE Enhanced Security from the server.

- 1 In IE, navigate to Tools — Internet Options — Security.

- 2 Select **Internet Security** and click **Custom Level**.
- 3 In the **Security Settings** dialog box, configure these options:
 - *ActiveX controls and plug-ins*
 - Download Signed ActiveX Controls (Prompt)
 - Run ActiveX controls and plug-ins (Enable)
 - Script ActiveX controls marked safe for scripting (Enable)
 - *Downloads*
 - File Download (Enable)
 - *Miscellaneous*
 - Allow Meta-Refresh (Enable)
 - Display Mixed Content (Enable)
 - Navigate sub-frames across different domains (Enable)
 - Submit non-encrypted form data (Enable)
 - Userdata persistence (Enable)
 - *Scripting*
 - Active Scripting (Enable)
- 4 Click **OK**.
- 5 On the **Privacy** tab, select these settings:
 - Privacy Level (Medium)
 - Turn off Pop-up blocking
- 6 On the **Content** tab, turn off all "auto complete" options
- 7 On the **Advanced** tab, select these options:
 - *Browsing*
 - Disable script debugging (ON)
 - Enable Page Transitions (OFF)
 - *Security*
 - Warn if changing between secure and non secure mode (OFF)
 - Warn if forms submittal is being redirected (OFF)
 - Warn about invalid site certificates (OFF)

- Empty Temporary Internet Files folder when browser is closed (ON)

Recommended Settings for IIS 6.0 “Web Garden”

In Windows 2003 (IIS 6.0), there are application pool settings that may cripple navigation in WebXM.

Ensure that the maximum number of worker processes (Web Garden setting) is set to 1. If raised to higher than 1, you may experience inconsistent navigation through the webapp. For example, when you are configuring a content scan job the navigation through the job properties may jump to the wrong screen and/or the setting you enter may disappear.

Win2003 and SQL

There is a web edition of Windows 2003 Server that will **not** allow SQL Server 2000 to be installed on it. Ensure the correct version of Windows 2003 Server is installed before proceeding with your deployment.

Moving an Oracle Database

If you need to move a WebXM database from one Oracle server to another, you can use the Export/Import utilities available in Oracle. The Export utility exports the tablespace to a file, and the Import utility imports the file to the new server.

Q & A

Will WebXM slow my web server performance while it's running?

Watchfire adheres to the HTTP 1.1 standard; therefore, the simple answer is NO.

WebXM should not significantly degrade web server performance. WebXM and WebXM Desktop should not be deployed on the web server itself, but on a separate machine that can view the website and perform a scan from there. The reason it does not degrade performance is based on the fact that, just as any other visitor or crawler crawls the website, the web server sees this as only one user visiting the website. However, if you experience slow response times, you can limit the number of links that processes simultaneously or schedule WebXM and WebXM Desktop to check the site after peak operating hours.

Will WebXM affect my traffic log file hits?

WebXM will increase your hits because it will request every URL on the entire site. However, the run should be recorded as only one visit to your site. Also, many usage analysis tools enable you to filter out these hits so as not to affect your traffic counts. WebXM's analytics job provides the Blocked IP Addresses property to remove such hits. You can also change the user agent that you are using when scanning with WebXM so that you can easily filter it from your traffic statistics.

Can WebXM work through a Proxy server?

Yes. You need to specify the address of your proxy server in the Proxies Options window in Internet Explorer.

To define proxy server settings:

- 1 Open the Internet Properties dialog box.
- 2 Click on the Connections tab.
- 3 Click the **LAN Settings** button.
- 4 In the Local Area Network (LAN) Settings dialog box, select the **use a proxy server...** check box
- 5 Enter the **Address** and **Port** number of your proxy server, and then click **OK**.

Does WebXM support authenticated pages and secure socket layer?

Yes. To access password-protected pages you need to configure your user name and password in the Connection Options window. Secure Socket

Layer, commonly referred to as SSL, is a form of communication in an HTTP environment – when entering a secure area the HTTP will turn to an HTTPS, showing that there is encryption in use.

How does the database server affect WebXM performance?

Configuring a server class machine for the database server can result in a significant improvement in the number of links checked per minute. However, a poorly performing database machine will adversely affect the performance of the content agent.

How does the size of the website affect scan performance?

Scanning performance (measured in links checked per minute) tends to diminish as the size of the website increases.

What is the best scanning strategy? One large scan or several smaller scans?

When scanning large websites, consider breaking the site up into smaller chunks and scanning each one separately.

If you have a very powerful database server, consider breaking the scan up into a number of small jobs and running them simultaneously.

Can the download of WebXM Desktop be disabled?

WebXM Server provides users with the ability to download and install WebXM Desktop by clicking on the **Download WebXM Desktop** button on the User Properties page in the Administration Center. You can disable this functionality by deleting the WebXM Desktop setup file from the WebXM server. In doing so the Download WebXM Desktop button is removed from the user interface.

To remove the WebXM Desktop setup file, navigate to *<WebXM install folder>\Webapp* and delete *WebXM_DesktopSetup.exe*.

Can I install my agents outside the LAN?

The agents should not be installed outside your LAN because they probably won't be able to see the database behind the firewall.

Troubleshooting WebXM

I can't access the WebXM database

If you are not able to access the database, it may be in recovery mode.

Why aren't the reports I created for my jobs appearing in the report pack?

In order to see the reports in the report pack in your webspace, you must add them in the report pack properties.

For more information, refer to the *WebXM User Guide*.

Why did my content scan stop or get suspended?

There are a number of reasons why a content scan stops:

- When the page limit parameter for your installation of WebXM is reached, the scan stops and an entry is made in the job's log
- Network problems that cause the connections between the WebXM Control Center, agent, and database to be lost
- The database server ran out of memory

For more information about the SQL Server memory settings, see "Memory" on page 5-5.

None of the scans seem to be doing anything; how can I find out what's happening?

If you have scan jobs that seem to be stalled, you can use a SQL database query to examine the data about the database connections.

The results of this query will include a column, called "spid" and a column called "blocked". If a row, representing a connection, contains a non-zero value in the "blocked" column, then the connection is being blocked.

For example, in Table 6-1, the database connection with spid = 64 is blocked by the connection with spid = 58. The connection with spid = 58 is not blocked.

	spid	blocked
6	58	0
7	60	0
8	61	0

Table 6-1 Blocked SQL Connection Example

	spid	blocked
9	62	0
10	64	0
11	64	64
12	64	58

Table 6-1 Blocked SQL Connection Example (continued)

Error message when attempting to publish or get projects between WebXM and WebXM Desktop

Attempts to publish or get projects generate the following error message:

An unknown error occurred while enumerating the jobs on the server.

(Details: "-2146827864:Object required")

This happens when there are jobs with "NULL" users as creators and last modifiers in the WebXM database. The user account that created or last modified the job was subsequently deleted from WebXM. You can correct this by editing the *wdrequest.asp* file, as follows:

After:

```
----script starts here ----
```

```
{
var                                oJob;
oJob                              = oJobColl.Job(i);
oUser                            = oJob.LastModifiedUser
----script ends here ----
```

Add:

```
----script starts here ----
```

```
if(oUser == null)
{
jr.AddJob(oJob.ID, oJob.PropsModifiedDateTime, "",
oJob.Description,
oJob.Contact,oJob.Name);
}
else
{
jr.AddJob(oJob.ID,
oJob.PropsModifiedDateTime,
oUser.FullName, oJob.Description,
oJob.Contact,oJob.Name);
}
```

```
}  
----script ends here ----  
(“----scripts... ----” indicate where the scripts start and end.)
```

Error message when attempting to run a scan job on WebXM via the browser

Attempts to access WebXM, via a web browser, generate the following error message:

Description: The remote server machine does not exist or is unavailable

Number: 800a01ce

To correct this problem:

- 1 Open the IIS Internet Services Manager.
- 2 Expand the computer icon, and then expand Default Web Site.
- 3 Right-click on the WebXM virtual directory, and then select **Properties**.
- 4 On the Virtual Directory tab, click the **Unload** button.
- 5 Click **OK**, and then close the IIS Internet Services Manager.
- 6 Close the browsers, and then reopen WebXM.

I upgraded WebXM Desktop and now my scan's status remains as "Starting"

You can correct this problem by editing the registry. In the Windows Registry, navigate to *HKEY_CURRENT_USER\Software\Watchfire\WebXM\ContentXM\Network Connection* and delete all the values except "Default".

Error message that log file for database is full.

Receives message "The log file for database WebXM is full. Backup the transaction log for the database to free up some log space --- dberror : 0x80040e14 WebXM error 0x8004070f"

To correct this problem, establish a database maintenance plan that includes a process to back up your database regularly.

Index

A

- Accessibility module 1-2
- agent 1-6, 3-12
 - collection 4-3
 - content 3-10
 - infrastructure 3-10
 - installation 3-12
 - interaction 3-10
 - operating systems 2-7, 2-8
 - processing 4-3
 - session 4-3
- Agent server 5-13
 - multiple instances 2-18
- agent service 1-6
 - operating systems 2-7, 2-8
 - start 3-17, 6-4
 - verify 6-4
- alerting service 1-7, 3-10
 - start 3-17, 6-4
 - verify 6-4
- Analytics
 - architecture 4-2
 - data tags, syntax 4-6
 - deploying 4-5, 4-20
 - JavaScript syntax 4-15
 - load balancing 4-6
 - module 1-2
- authentication 5-3, 5-6
- auto create statistics 5-7
- auto update statistics 5-7

B

- backup 5-16
 - database recovery 5-9
 - SQL Server 5-8

strategy 5-8, 5-16

browsers

- WebXM Desktop 2-10
- WebXM Server 2-6

C

- checklist, preinstall 3-2
- collection agent 4-3
- configuration
 - database 5-15
 - database file 5-8
 - Internet Explorer 6-5
 - modify 3-17
 - SQL log file 5-8
 - WebXM components 3-14, 3-17
 - WebXM Server 3-10, 3-12
- Configuration Wizard 3-10, 3-12, 3-17
- connect to a database 1-6, 3-15, 3-17
- contacting
 - Watchfire 1-9
- Content 1-7, 1-8
- content agent 3-10
 - installation 3-12
- create a database 1-6, 3-15, 3-17
 - tablespace 5-13
- custom options 6-2
- Customer Portal 1-8

D

- data tags
 - automatic 4-17
 - character encoding 4-16

- collection process 4-3
 - implementing 4-17
 - syntax 4-6
- database 5-2
 - backup 5-8, 5-16
 - configuration 5-5
 - connect to existing 1-6, 3-15, 3-17
 - create new 1-6, 3-15, 3-17
 - estimate size 5-2
 - file configuration 5-8
 - incompatible version 3-15
 - maintenance 5-8, 5-9, 5-16
 - Microsoft SQL Server
 - see* SQL Server
 - monitoring performance 5-9, 5-17
 - moving 6-7
 - MSDE 3-19
 - NetGenesis 3-18
 - options 5-7
 - Oracle 3-15, 5-11
 - permissions 5-3
 - properties 5-5
 - recovery 5-9, 5-16
 - security 5-3
 - shrink 5-9
 - space allocation 5-15
 - SQL Server 3-15, 5-4
 - statistics 5-7
 - tables 5-4, 5-11
- Database Properties 5-5, 5-7
 - Auto create statistics 5-7
 - Auto update statistics 5-7
 - Recovery Model 5-7

- default language 5-6
- defragment 5-9, 5-17
- dn parameter 4-9
- downloadTime parameter 4-10

E

- error messages
 - scanning 6-12

F

- Flash 3-11
- force HTTP 1.0 6-5
- foreign key constraints 5-5, 5-12
- full-text search indexing 5-7

I

- IIS 6.0, recommended settings 6-7
- infrastructure agent 3-10
- installation
 - additional agents 3-12
 - checklist 3-2
 - MSDE 3-19
 - multiple instances of
 - WebXM 2-16, 3-13
 - multiple servers 2-13, 2-14, 3-10, 3-11
 - Oracle database 5-13
 - preparation 3-7, 3-8
 - required information 3-6
 - single server 2-13, 2-14, 3-9
 - WebXM 3-8
 - WebXM Desktop 3-20
 - workflow 3-2, 3-21
- instances 2-16, 3-13, 4-6
- integrated authentication 3-16, 3-22
- Interaction 1-7, 1-8
- interaction agent 3-10
 - installation 3-12
- International website scanning 3-7
- Internet Explorer
 - configuring for advanced login 6-5
 - proxy settings 6-8
- Inventory module 1-2

J

- JavaScript syntax 4-15

L

- languages, scanning 3-7
- license key 3-17
- Listener 5-13
- log file
 - SQL 5-8

M

- maintenance
 - SQL Server 5-8
- memory 5-5
 - dynamically configure 5-6
- Microsoft SQL Server
 - see* SQL Server
- monitoring performance
 - CPU usage 5-10, 5-17
 - processes to monitor 5-10, 5-17
- MSDE 1-6
 - installation 3-19
- multiple instances 2-16
 - Agent 2-18
 - WebXM Control Center 2-17
- multiple servers 2-13, 2-14
 - installation 3-10, 3-11

N

- nested triggers 5-6
- Net Service 5-13
- NetGenesis
 - configuring for Traffic module 3-18

O

- operating systems
 - agent 2-7, 2-8
 - agent service 2-7, 2-8
 - WebXM Desktop 2-10
 - WebXM Server 2-6
- options.xml 6-2
- Oracle 3-15, 5-11
 - client requirements 5-14
 - create

- Listener 5-13
- Net Service 5-13
 - user 5-14

- DBAStudio 5-18
- maintain database 5-16
- monitoring performance 5-17
- moving a database 6-7
- packages 5-12
- server installation 5-13
- SQLPlus 5-18
- table space 5-15
- tables 5-11
- troubleshooting 5-18
- user permissions 5-14

- Oracle Client 5-13
- Oracle Provider 2-8, 3-15, 5-13, 5-14

P

- passwords 2-11
- path parameter 4-8
- Perfmon 5-9, 5-10, 5-17, 5-18
- performance
 - CPU configuration 6-3
 - full-text search 5-7
 - hardware configuration 6-2, 6-3
 - JavaScript 6-4
 - latency 6-3
 - monitoring 5-9, 5-17
 - multiple instances of
 - WebXM 3-13
 - report generation 6-3
 - size of scan 6-3
- planning 2-2, 2-12
 - WebXM Desktop 2-4
 - WebXM Server 2-2
- planning issues
 - corporate standards 2-12
 - multiple servers 2-13
 - organizational structure 2-12
 - performance 2-13
 - physical environment 2-13
 - single server 2-13
 - website structure 2-12

- pre-authorized authentication 3-16, 3-22
- Privacy module 1-2
 - activation 3-17
- process blocking 5-11
- processing agent 4-3
- proxy settings 2-13, 6-8

Q

- Quality module 1-2
- Query Analyzer 5-10
- query governor 5-7

R

- recovery model 5-7
- referential integrity 5-5, 5-12
- report generation 6-3
- restore database 5-16

S

- scan performance
 - collecting accessibility data 2-3, 2-4
- Security module 1-2
- server settings 5-6
- service account 5-3
 - local administrator rights 3-8
 - passwords 2-11
- session agent 4-3
- shrinking database 5-9
- single server 2-13, 2-14
 - installation 3-9
- SiteMinder
 - pre-authorized authentication 3-16, 3-22
 - WebXM Desktop and 3-23
 - WebXM integration and 3-22
- space allocation 5-15
- SQL Server 3-15, 5-4
 - authentication 5-6
 - configuration 5-5

- database size 5-2
- default language 5-6
- log file configuration 5-8
- monitoring performance 5-9
- nested triggers 5-6
- process blocking 5-11
- properties 5-5
- query governor 5-7
- shrink database 5-9
- troubleshooting 5-10
- SQL Server Enterprise Manager 5-10
- SQL Server Properties 5-5
 - Memory 5-5
 - Processor 5-6
 - Security 5-6
 - Server Settings 5-6
- stored procedures 5-4
- support
 - technical 1-8
- system requirements
 - WebXM database 2-8
 - WebXM Desktop 2-9
 - WebXM Server 2-5

T

- tables 5-4, 5-11
- tablespace 5-13, 5-15
 - create new 5-13
 - user permissions 5-14
- technical support 1-8
- terminology 1-3
- traffic analysis 6-8
- Traffic module 1-2
 - activation 3-17
 - configuration for NetGenesis 3-18
- troubleshooting 5-18, 6-2
 - common problems 5-10
 - DBAStudio 5-18
 - process blocking 5-11

- Query Analyzer 5-10
- SQL Server 5-10
- SQL Server Enterprise Manager 5-10
- SQL Server issues 5-10
- SQLPlus 5-18
- tools 5-10, 5-18
- Windows Events Log 5-10, 5-18
- Windows Performance Monitor 5-10, 5-18

U

- upgrading
 - checklist 3-21
 - configure components 3-14
- user roles 5-3

W

- Watchfire
 - contacting 1-9
- Web Garden, recommended settings 6-7
- WebXM
 - components 1-5
 - database 1-6, 5-2
 - installation 3-8
 - multiple instances 3-13
 - planning 2-2
 - terminology 1-3
- WebXM Control Center 1-5, 2-6, 3-10
 - multiple instances 2-17
- WebXM Desktop 1-3
 - browsers 2-10
 - download 3-20
 - hardware requirements 2-9
 - installation 3-20
 - operating systems 2-10
 - planning 2-4
 - preparation 3-8

WebXM Server

- browsers 2-6

- configuration 3-10, 3-12,
3-14

- hardware requirements 2-5

- operating systems 2-6

- planning 2-2

- preparation 3-7

Win2003 and SQL issues 6-7

Windows Events Log 5-10, 5-18

Windows Performance Monitor
5-9, 5-10, 5-17, 5-18