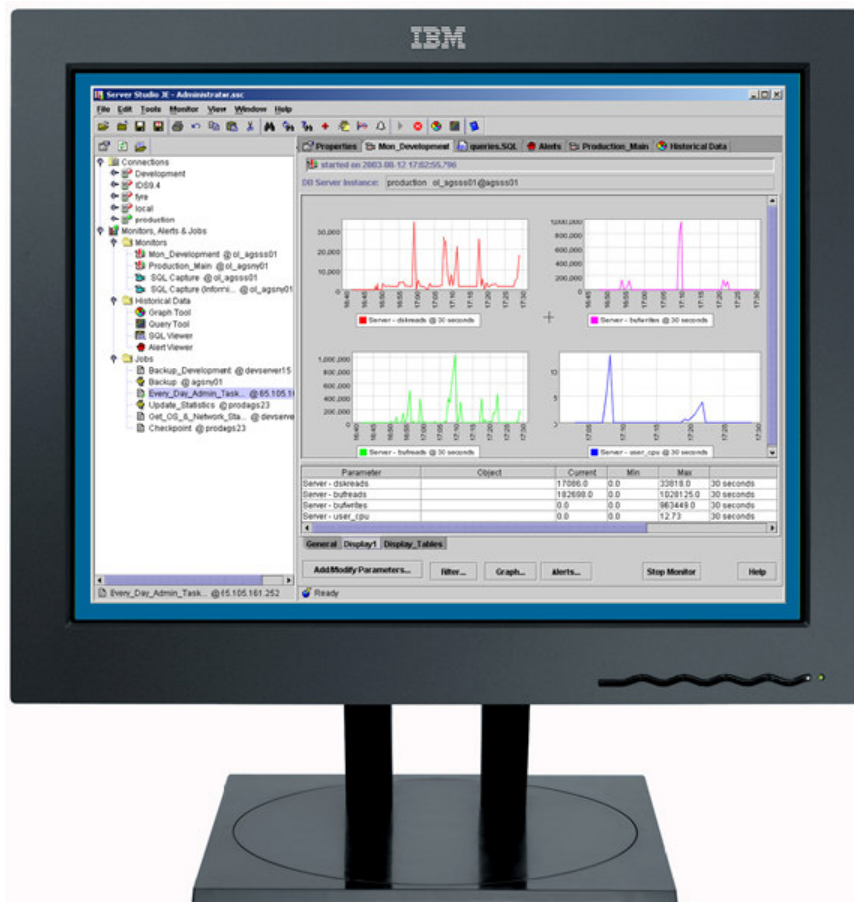




Sentinel™

Quick Start Guide



Performance Monitoring, Optimization, Automation and
Database Schema Change Management
for IBM Informix® Dynamic Server

Abstract

Sentinel™ automates proactive 24 × 7 operational monitoring of IBM Informix® Dynamic Servers, provides an extensive array of sophisticated diagnostics tools to tune critical performance parameters, executes regular database maintenance tasks, enables setting of highly flexible multi-level alerts to flag availability and performance degradation problems and manages autonomic responses to server events in real time. Sentinel™, together with Server Studio™, provide wide-ranging functionality and powerful infrastructure to support vital IBM Informix® databases — from initial design, development and testing, all the way through to production deployment and successful operational service.

In addition, facilities provided by Sentinel — Change Management Option streamline database configuration change management by simplifying the tedious and error-prone task of maintaining traceable records of a database schema state changes and automatically notifying administrators when a change from a baseline schema is detected. Using a scheduled schema audit job, you can compare periodically a single 'baseline' database schema, for example, a snapshot of a tested QA database, against multiple production databases that can reside on different Informix servers. This job will detect any modifications made in a production schema, saving you a lot of time and efforts when you need to keep multiple production server instances synchronized with a single source database schema. Change Management subsystem allows to maintain traceable record of database security states, such as user's and role's permissions and track security changes for audit purposes. It also enables capturing periodic database schema version control snapshots ("VCS") and retaining them in the built-in repository for easy roll-out, roll-back and recovery of lost or damaged database objects.

This guide introduces database administrators and developers unfamiliar with Sentinel™ to how monitors and jobs are set up and organized, and how performance data are used to provide optimization and event response automation for IBM Informix® Dynamic Server.

©2014 Advanced Global Systems, Ltd. All rights reserved. The information contained in this document represents the current view of AGS on the issues discussed as of the date of publication. Because AGS must respond to changing market conditions, it should not be interpreted to be a commitment on the part of AGS, and AGS cannot guarantee the accuracy of any information presented after the date of publication.

This guide is for informational purposes only. AGS MAKES NO WARRANTIES, EXPRESS OR IMPLIED, IN THIS DOCUMENT.

AGS, Server Studio and Sentinel are either registered trademarks or trademarks of Advanced Global Systems, Ltd. in the United States and/or other countries.

Other product and company names mentioned herein may be the trademarks of their respective owners.

Advanced Global Systems, Ltd. • One West Street, Suite 100 • New York, NY 10004 • USA
01/07

CONTENTS

INTRODUCTION	1
SENTINEL™	3
Architecture	3
Installation	5
Configuration	5
Connection Panel	5
Administration Panel	6
Sentinel Server User Account Credentials	6
Sentinel Configuration Control Policies	6
Master Database Login Credentials Dialog Box	7
Repositories Panel	7
Cloudscape Repository Options	8
Informix Repository Options	8
Options Panel	8
Monitoring Blackouts Panel	9
Adding a new blackout period	9
Editing an existing blackout period	9
Configuring blackout periods notifications	9
Starting and Stopping Sentinel Service	10
Microsoft Windows	10
UNIX and LINUX	10
Troubleshooting	11
Problems starting Sentinel server for a first time	11
Windows	11
UNIX and Linux	11
Problems starting Sentinel server after it successfully worked for a period of time	12
SQL Capture issues	12
Debugging Mode	13
MONITORS.....	14
Creating Monitors	15
Selecting Informix Performance Parameters	15
Selecting OS Performance Parameters	15
Defining Filters	17
Defining Alerts	17
Defining Real-Time Graphs	18
Editing Monitors Configuration	20
Replicating Monitors	20
Deleting Monitors	20
Running Monitors	20
Creating New Sentinel Connection	21
ALERTS.....	22
User-Defined Alerts	22

Creating User-Defined Alerts	23
Jobs Alerts	24
Schema Audit Jobs Alerts	25
System Alerts	25
Viewing Alert Events	26
JOBS	27
Configuring Jobs	28
Defining “OS Command”-level Jobs	28
Defining SQL Script/Stored Procedure Jobs	29
SYSTEM Statements	29
Specifying Environment Variables in SYSTEM Statements	30
Examples of the SYSTEM Statement: UNIX	30
Examples of the SYSTEM Statement: Windows NT	31
Using Jobs	31
Manual Job Execution	31
Scheduled Job Execution	32
Job Execution in Response to Alert Event	32
Editing Jobs	32
Deleting Jobs	32
SQL CAPTURE	34
Creating SQL Capture Process	34
Viewing Captured SQL Statements	35
PERFORMANCE HISTORY REPOSITORY	37
Viewing Historical Performance Statistics	37
Graphical Tool	37
Query Tool	40
Repository Structure	40
LOAD TESTING WITH BENCHMARK RUNNER™	41
Creating Benchmark Runner™ Scenarios	41
Using Variables in Benchmark Runner™ Scenarios	42
Editing Benchmark Runner™ Scenarios	43
CHANGE MANAGEMENT	44
Change Management Repository	44
Schema Audit Jobs	44
Creating Schema Audit Job	45
Editing Schema Audit Job	46
Deleting Schema Audit Job	47
Viewing Schema Audit Reports	47
Version Snapshots	47
Creating Version Snapshots	48
Creating snapshot for a single database	48

Creating a snapshot for individual database object	48
Creating a snapshot for a group of databases	49
Using Version Snapshots	49
Viewing Snapshot Content	49
Generating SQL script based on a snapshot	49
Comparing Snapshot with a current database state	49
Comparing Snapshot with another snapshot	49
Deleting Version Snapshots	49
Deleting a group of selected version snapshots	50
Deleting all snapshots for Informix server instance	50

APPENDIX A: ADDITIONAL INFORMATION.....51

APPENDIX B:52

INTRODUCTION

A great part of managing IBM Informix® Dynamic Server involves performing regular administration and maintenance tasks, responding to unexpected system events and keeping your database server tuned for optimal performance. This is easier said than done. To ease day-to-day monitoring and administration of your entire IDS infrastructure, Sentinel™ provides a powerful set of seamlessly integrated, state-of-the-art facilities that play a pivotal role in providing comprehensive performance monitoring, tuning and event response automation for IBM Informix® Dynamic Servers infrastructure.

Performance of your database server, to a great degree, depends on the kinds of applications that you run on your system. The causes of performance problems on OLTP (online transaction processing) systems are very different from the causes of problems on systems that are used primarily for DSS (decision support system) query applications. Systems with mixed use provide an even greater performance-tuning challenge and require a more sophisticated analysis of performance-problem causes. Some configuration adjustments can unintentionally decrease performance or cause other negative side effects. Before you begin to adjust the configuration of your database server, it is vital to evaluate the performance history and profile of your current system configuration. Equally, when you do make configuration adjustments to your system, it is important to measure objectively the effects of such changes on the performance of your database server.

Sentinel™ enables you to monitor in real time resource use of your database server, ongoing transactions and query performance. It also accumulates detailed time-series data on performance history and profile of your system's resource-utilization. For example, you can track the CPU utilization, paging-out rate and the I/O transfer rates for the various disks on your system, to identify peak-use levels, peak-use intervals and heavily loaded resources. Similarly, you can monitor fragment use, to determine whether your fragmentation scheme is correctly configured. If your system's throughput and response times suddenly degrade, you can capture and analyze SQL statements executed by users to determine if particular queries are the cause of inadequate performance. To customize monitoring solution for your system's resource use that are appropriate for your database server configuration and the applications that run on it, Sentinel™ gives you the ability to define your own monitors from a matrix of nearly 160 IDS and host's platform performance parameters.

With detail performance history and profile of your system's resource-utilization information on hand, you can begin to track the cause of problems as soon as users report slow response or inadequate throughput. If history is not available, you must start tracking performance after a problem arises and you might not be able to tell when and how the problem began. Trying to identify problems after the fact significantly delays resolution of performance problems.

Ability to track the cause of performance problems and correct them quickly is good; however, the ability to prevent them from occurring is even more desirable. It is better to be proactive than reactive. Sentinel™ enables you to create multi-level

alerts that notify you of impending performance problems or critical events your database server encounters. Moreover, you can create customized autonomic responses that are triggered by such alerts and execute a series of OS commands, SQL scripts and IDS native utilities. For example, if an 80%-full threshold alert is generated by the dbspace monitor, a custom job script can be automatically triggered to add an additional chunk to it. Regularly scheduled preventive maintenance tasks, such as backing up your databases, can be also automated.

This guide describes effective use of Sentinel monitors, alarms, load scenarios, custom jobs, scheduling and autonomic event response management from the perspective of IBM Informix® Dynamic Server performance optimization. It also covers Database Schema Change Management facilities that include schema audit jobs, schema audit reports and alerts, and schema version snapshots.

The following sections cover these topics:

- **Sentinel Server**
- **Monitors**
- **Alerts**
- **Jobs**
- **SQL Capture**
- **Performance Statistics Repository**
- **Load Testing with Benchmark Runner™**
- **Change Management**

For topics on use of Server Studio™ not covered in this guide, see the section “For More Information” at the end of this document.

SENTINEL™

Architecture

Sentinel™ implements an agentless client-server architecture to provide comprehensive performance monitoring, tuning and event response automation for IBM Informix® Dynamic Server. It consists of two major components, the Sentinel™ server ("Sentinel") and the graphical user interface client which resides in Server Studio™ ("Sentinel Console"). To assure system security, Sentinel server and its Console components communicate with each other by encrypted messages over secure TCP/IP protocol.

Sentinel server runs as a service on Windows platforms or as a background daemon process on UNIX and LINUX platforms. It is responsible for providing performance monitoring, data collection, event response automation and scheduled task execution functionality for one or multiple IBM Informix® Dynamic Server instances. While it is possible to install Sentinel on the same host computer where the monitored IDS instance is deployed, it is recommended to install it on the different host, to minimize the impact on the resources of the database server host.

Correspondingly, each Sentinel Console provides user interaction required to manage one or multiple Sentinel server instances deployed in your enterprise. Essentially, as depicted schematically in the diagram below, Sentinel implements a many-to-many architecture, where each Sentinel server can service one or multiple IDS instances and, in turn, each Sentinel Console can service one or multiple Sentinel servers.



To enable monitoring of an IDS instance, you need to create one or more monitors for it. A monitor is a collection of server performance parameter sensors and user-defined alert conditions related to those parameters. Multiple logical monitors for a given IDS instance can be created and function concurrently. Sentinel internally aggregates and optimizes performance sensors' data requests from multiple logical monitors to minimize the impact on the target IDS instance.

Sentinel is responsible for collecting monitored parameters from one or multiple IDS instances, generating user-defined alert notifications in real-time and executing regularly scheduled tasks or alert-triggered event responses. It utilizes either an embedded SQL-compliant database or any available IDS instance database as a repository for time-series performance data and alert events collected from all monitored IDS instances for performance history analysis and profiling of your system's resource-utilization. You can choose which parameters data should be stored in the Sentinel repository when monitors are defined.

Sentinel Console is responsible for user interaction and provides the necessary graphical tools for the Sentinel server configuration, creation and management of monitors, alerts and job definitions. It also provides you with real-time performance graphs and contains the required tools to enable analysis of historical performance and resource-utilization data collected by Sentinel server.

Some of the key Sentinel Console's functions are:

- Connecting to and configuring Sentinel servers
- Starting and stopping Sentinel servers
- Creating and modifying monitors definitions
- Starting and stopping monitors
- Displaying real-time graphs of monitored IDS or host platform parameters
- Creating and modifying definitions of jobs
- Manually executing jobs
- Scheduling jobs
- Viewing alert events generated by monitors and jobs
- Viewing job execution history
- Viewing SQL statements collected by SQL Capture processes
- Displaying status of currently running monitors and jobs
- Viewing graphical representation of historical data collected by Sentinel
- Querying and exporting historical data collected in the Sentinel's repository

Installation

Sentinel server is installed as part of the full Server Studio installation and, as a result, it is always available at the same workstation where the Server Studio client is installed. This makes it easier to configure and test entire Sentinel locally before transferring the server to a dedicated host in the production environment. There is also separate installation process for the Sentinel server that allows installing it in a console or command-line mode. See readme.html, for the complete Server Studio installation instructions.

Configuration

To access Sentinel server, you have to create a connection to it from your Server Studio client. Once you connect to a running Sentinel server, you can manage it: change its properties, create monitors and alert definitions, access Sentinel performance repository, define jobs, and perform any other Sentinel related functions.

Sentinel Configuration dialog box consists of five tabs:

- Connection Panel
- Repositories Panel
- Options Panel
- Administration Panel
- Monitoring Blackouts

When creating a new Sentinel server connection or when the selected Sentinel server connection is not established only the Connection Panel is available. Other panels become accessible only when a connection to the Sentinel server is established. In addition, Administration Panel is available only when you connected to the Sentinel server using Sentinel Administrator login credentials.

Connection Panel

Name – define a logical name for the Sentinel server connection. For example, 'Sentinel for New York IDS Cluster' or 'Sentinel for London Office Network'

Host - name or IP address of workstation on which Sentinel server runs

Port - port number on which Sentinel server runs. Default port number for Sentinel is 10556 but it can be changed using sentinel.cfg file that is located in <SENTINEL_ROOT>\sentinel\config subdirectory. After the Sentinel port number is changed, Sentinel server has to be restarted for the change to take effect. When connecting to an already running Sentinel server, the port number in this field should match the port number in sentinel.cfg configuration file.

Status – this 'display only' field shows the status of Sentinel server after a connection from the Server Studio client has been established. It displays N/A if a Sentinel connection is not established.

Auto Connect – Select this checkbox if you want Server Studio client automatically establish a connection to the selected Sentinel server, when Server Studio application starts. When this option is cleared, a manual connection has to be established from the Sentinel Explorer tree (Sentinel-Automation tab in Explorer panel)

Sentinel Server Login Credentials – by default the Sentinel Server has a single user that has Administrator role:

User Name: **sentinel**

Password: **sentinel**

After the first login to a Sentinel server as Administrator, you can change the password for sentinel account as well as create additional Sentinel users using Administration tab of Sentinel Configuration Dialog box. The Sentinel login credentials are required because multiple users/DBAs can connect to Sentinel over the TCP/IP network and you might want to restrict access only to authorized users and limit their ability to change Sentinel server configuration parameters.

Administration Panel

This panel is available only to the user with the Administrator role. By default this is a user with a following Sentinel login credentials:

User Name: **sentinel**

Password: **sentinel**

Sentinel Server User Account Credentials

Once you login as Administrator, you can change this default administrator login by selecting it in the list of Sentinel accounts and pressing the Edit button.

To add a non-administrator Sentinel user, press the New button. Enter a new user name and password and optionally select 'Change configuration' in the permission grid. If this checkbox is selected, this user will be able to change the location of the Sentinel repositories, SMTP settings and other Sentinel options. However, creation of new Sentinel user accounts is reserved to the Administrator account.

Administrator account is also allowed to edit monitors and jobs created by different Sentinel users. Other Sentinel users can edit only their own monitors and jobs.

Sentinel Configuration Control Policies

The functionality in this panel allows for creation of Informix database servers 'master logins'. Sentinel server runs monitors and jobs created by users using its own JDBC and Telnet/SSH connection. Because multiple users can create monitors, alerts and job definitions there might be a need to force these processes to use a single 'master login' instead of Informix logins supplied by users who create

Sentinel monitors. By default, when DBAs create a Sentinel monitor, they have to enter the Informix IDS login to run this monitor. Because Sentinel monitors usually run 24x7 in 'unattended' mode and monitor login information is stored on Sentinel server as part of monitor definition, you might want to avoid using personal DBAs full privileged logins just for monitor running purposes. In this case, you can create a special Informix login – master login that has limited Informix privileges - just for running monitors and use this login for all newly created monitors and jobs. Sentinel Administrator creates a 'master login' that only Sentinel Administrator knows but other Sentinel users just 'blindly' use when creating new monitor or job definition instead of their own personal Informix logins.

To create a 'master login', press the New button.

NOTE Master Logins are forced only when 'Require the use of master database server login credentials for creation of new monitors' checkbox is selected. Otherwise, Sentinel monitors use login credentials of Server Studio users that create a monitor or job.

Master Database Login Credentials Dialog Box

Master Login Identifier – enter the logical name for this database credential set. For example, 'sentinel_runner'. Sentinel users will see this name in the list when selecting login credentials during monitor or job creation process.

Database Server Connection – specify the server name and host name for Informix IDS server, for which you create this master login.

Database Server Login Credentials – specify the Informix user name and password that this master login will use to connect to the Informix server.

OS Host Login Credentials – specify the login name and the authorization method for your host access protocol (telnet or SSH). These parameters are needed only if Sentinel users will create monitors that require OS access, such as OS CPU Busy, or Informix Checkpoint Duration parameter. This is also required when Sentinel users create Sentinel jobs defined as 'Remote OS Command'. For example, when a job, such as Informix Backup, is scheduled to run periodically, Sentinel server will use the provided OS master login to run the specified database backup script rather than the login information originally supplied with the job.

Repositories Panel

Repository Type – by default Sentinel server uses the embedded Cloudscape database as a repository. Optionally you can use an external Informix database as a repository instead of the embedded Cloudscape database. In this case, the database size is only limited by available space in Informix dbspace in which you create a repository database.

Cloudscape Repository Options

Database path – path to the embedded repository directory

Maximum Database Size – specifies the maximum repository size in Mb. This option allows to prevent excessive growth of the repository database on the hard drive.

Informix Repository Options

Connection – your Server Studio connection that points to the Informix server that will serve as the Sentinel repository. Press “...” button to create a connection if you do not have it yet.

Create in DB Space – choose the DB Space where the 'sentinel' repository database will be created.

Table Extent Size - specifies the extent size for repository tables. If you expect that the repository will be large choose larger extent size.

Options Panel

SMTP Server - In order to send user-defined alert notifications via email, pager or mobile phone, an SMTP server that will process email messages has to be specified. SMTP server can reside on any workstation but it should allow relay of email messages from the Sentinel host workstation. If the Sentinel server is installed under Windows Server, you can use Windows built-in "Simple Mail Transport Protocol" service.

Memory - the maximum amount of memory allowed for the Sentinel server process. Increase this parameter to be ¼ (maximum of 1GB) of the installed physical RAM to improve performance and data capacity of the server process. It's recommended that you set this to the maximum allowable size when you are planning to work with a large number of monitored Informix instances or monitor large number of individual objects, such as tables or chunks. It also has to be increased if your Sentinel repository database growth fast. This parameter can be changed from the Server Studio UI or directly modified in the sentinel.cfg file in the <AGS>\sentinel\conf subdirectory. The change takes effect only after you restart the Sentinel server.

Admin Email Address - The specified administrator will receive Sentinel system alerts, such as notifications about monitor failures. If the administrator email is not provided, you can view system events using Server Studio's Alert Events Viewer in the Automation-Sentinel tab of the Object Explorer.

Monitoring Blackouts Panel

You can schedule alert blackout periods for planned Informix server down time to avoid receiving false alerts. This is useful when you know in advance that you are going to take Informix server down for maintenance or an upgrade. When an alert blackout period is activated, Sentinel will not send out any alerts for the specified servers. Sentinel administrators will be notified whenever a blackout period starts and ends via Server Studio alerts and email messages sent to administrator's email address specified in this window's Options tab.

Adding a new blackout period

Press the Add button to add a new blackout period for an Informix server or a group of servers. The Maintenance Blackout Definition dialog box will open. Enter a description of the maintenance blackout period in the Description dialog box. The description is used to identify a maintenance period in the list. Choose the interval type in the Occurs groupbox: Daily, Weekly or Once. For Daily and Weekly interval types, choose additional parameters, such as days of week on which the blackout interval happens. In the Duration groupbox, choose duration for a specified interval type. The Duration field specifies how long the blackout interval will be in effect. For Daily and Weekly interval types, you must also choose Start Date, End Date and Start Time parameters.

Use 'Apply to the following servers' grid to select Informix servers to which this maintenance period applies. NOTE: You will be able to edit this list at later time, if new Informix servers are added to which this maintenance period applies as well.

Press OK button to close Maintenance Blackout Definition dialog box and add the new maintenance period to the list.

Editing an existing blackout period

To edit an existing blackout period, select a blackout period in the list and press Edit button to open Maintenance Blackout Definition dialog box.

Configuring blackout periods notifications

Check 'Send Blackout Activation Alerts to Administrator Email' checkbox if you want to receive notifications after blackout periods begin and end. Otherwise, you can view those notifications in the Server Studio's Alerts Events panel in Sentinel-Automation tab.

Starting and Stopping Sentinel Service

The Sentinel server component is installed as a service under Windows 2000/XP and NT platforms. Under UNIX and Linux platforms, the server runs as a background process.

Microsoft Windows

- **Starting the Sentinel Server.** If Sentinel server is installed on the same workstation as Server Studio, in order to start Sentinel server, use Start Sentinel shortcut menu when the required Sentinel Server node is selected in the Sentinel-Automation Explorer. Otherwise, use Windows Services tool to start the 'AGS Sentinel Server' service. To open the Windows Services tool, use Windows Start → Programs → Administrative Tools → Services menu. You can switch the service start up type to Automatic, if you want to run the Sentinel server when this workstation is turned on. All active monitors, defined for this server, will start automatically as well. You will not have to open the Server Studio GUI to start the monitoring.
- **Stopping the Sentinel Server.** If Sentinel server is installed on the same workstation as Server Studio, in order to stop Sentinel server, use Stop Sentinel shortcut menu when the required Sentinel Server node is selected in the Sentinel-Automation Explorer. Otherwise, use Windows Services tool to stop the *AGS Sentinel Server* service. To open the Windows Services tool, use Windows Start → Programs → Administrative Tools → Services menu.

See Troubleshooting section if you have any problems starting or stopping Sentinel server.

UNIX and LINUX

If Sentinel server is installed on the same workstation as Server Studio, in order to start or stop Sentinel server, use Start Sentinel or Stop Sentinel shortcut menu when the required Sentinel Server node is selected in the Sentinel-Automation Explorer. Otherwise, execute from the command shell 'sentinel' launching script to start, stop and restart the server. It is located in the '<AGS>/sentinel' subdirectory.

- **Starting the Sentinel Server.** Use 'sentinel start' command to start the server.
- **Stopping the Sentinel Server.** Use 'sentinel stop' command to stop the server.
- **Restarting the Sentinel Server.** Use 'sentinel restart' command to restart the server.
- **Obtaining Sentinel status.** Use 'sentinel status' command to obtain the current Sentinel server status

Make sure that you have a proper execute permissions for this script and that the

AGS_HOME and JAVA_HOME environment variables defined in the beginning of this script point to correct directories for your installation. You can also include a call to the 'sentinel' command in the init.d script to start/stop the Sentinel server automatically during the OS start up and shutdown operations.

See Troubleshooting section if you have any problems starting or stopping Sentinel server.

Troubleshooting

Problems starting Sentinel server for a first time

If the Sentinel server does not start, the most common reason is that Java VM is not found by Sentinel launching process on your workstation. This might happen when Windows registry entries do not contain correct keys for default Java VM or Java VM was reinstalled after the Sentinel was installed on your workstation.

Windows

You can find the error log in:

1) Windows Event Viewer – the errors logged by sentinel.exe application. Use Windows Control Panel → Administrative Tools → Event Viewer to open Event Viewer, switch to Application Log section and look for messages from Source 'sentinel'.

2) *sentinel.log* text file created by Sentinel process in <AGS>\sentinel\logs subdirectory.

Errors appear in sentinel.log file only if the Sentinel service was actually started by Windows. If the service could not be started because of issues with Java VM or permissions issues, the error will be in the Windows Event Viewer.

To force Sentinel using a particular Java VM, create in <AGS>\sentinel subdirectory a text file named 'jvm' and in a first line of this file put a full path to 'jvm.dll' file from Java VM server or client subdirectories.

Example,

d:\java\jre1.6_25\bin\server\jvm.dll

UNIX and Linux

You can find the error log in 'sentinel.log' text file created by Sentinel process in <AGS>/sentinel/logs subdirectory.

Check *sentinel* script file to verify path specified in JAVA_HOME and AGS_HOME environment variables.

Problems starting Sentinel server after it successfully worked for a period of time

If you have problems starting Sentinel after it ran successfully for a period of time, first make sure that 'sentinel.exe' process does not run already on Windows workstation or 'java' Sentinel process from <AGS>/sentinel subdirectory does not run on UNIX/Linux. If normal commands to stop these processes do not work (Stop service command from Windows Services dialog box or 'sentinel stop' on UNIX/Linux) kill these processes using OS means. On Windows there is killmon.exe utility in <AGS>\sentinel subdirectory that can kill this OS process.

After the process is stopped, analyze the content of sentinel.log log file located in <AGS>\sentinel\logs subdirectory. It can provide the explanation of the potential problem with Sentinel. If Sentinel runs out of memory, increase Memory setting by editing configuration file sentinel.cfg in any text editor. You should increase the default maximum value of MEMORY parameter from a default of 256 to a value depending on what your monitoring tasks are and what is the size of the performance data repository. It makes sense to set it to 512 (use up to 512MB of the installed memory) if your workstation has enough RAM.

If the previous step does not help in restarting Sentinel server, rename 'state' subdirectory in <AGS>\sentinel to 'state_bak' and restart it again. If Sentinel still does not start, rename the Sentinel repository subdirectory (default is <AGS>\userdata\sentinel\stores but the actual path can be found in <AGS>\sentinel\config\sentinel.cfg file) and attempt to restart the Sentinel once more.

SQL Capture issues

SQL capture process works in a pooling mode retrieving opened prepared SQL statements (cursors) on a specified interval from Informix SMI tables. If you created SQL Capture process without specifying any filters, it retrieves all SQL statements prepared on the server. For large systems, there might be thousands and tens of thousands of opened SQL statements every moment of time and SQL Capture process will not be able to retrieve this data on the specified refresh interval.

If your SQL capture process does not return any data, you can run the following SELECT statement to estimate the number of SQL cursors currently opened on your system:

```
SELECT count(*) FROM sysmaster:sysconblock
```

If the count of opened cursors is high, set filters for SQL Capture process to limit retrieved SQL statements to only poorly performing statements (based on execution stats, such as estimated cost or number of sequential scans) or statements executed by specified users/hosts/sessions.

Debugging Mode

Sentinel provides a debugging mode, in which it writes a lot of additional information into the log file, that can help AGS support personnel to investigate and fix the issue.

To run Sentinel in a debugging mode:

- 1) Stop Sentinel server if it runs.
- 2) Edit <AGS>\sentinel\config\sentinel.cfg file in any text editor and edit DEBUG parameter to set it to TRUE :
DEBUG=TRUE
- 3) Restart the Sentinel server and try to reproduce the issue. After the issue is reproduced, look at the generated <AGS>\sentinel\logs\sentinel.log file to see if there are error messages in it explaining what the problem might be. If you can not find any clear error messages in the log file that explain the nature of the problem, please send us this log file along with the detailed description of the problem to:
support@serverstudio.com

MONITORS

A Sentinel server monitor is a named collection of Informix server performance parameters, related alert conditions, object filters and real-time graph definitions. The monitor is defined against a single IDS instance. You can select a subset from approximately 150 available performance parameters defined at server, table/index, dbspace, chunk, user session and OS host levels. For example, one monitor definition can contain a disk cache read percentage for the entire server and the number of requested locks for 10 most heavily used tables.

For each selected parameter you can specify its own refresh interval, set the flag whether the parameter's data should be stored in the historical database for future analysis, enter alert conditions, and optionally choose to display a real-time graph for it.

For table, chunk, space and session-level parameters, the data collection, alerting and graphing can be customized on database object level. For example, for a TABLE BUFFREADS parameter you might want to record data only for 5 tables, set alert condition on 100 tables and graph in real-time only 2 most critical tables.

Multiple monitors for the same database instance can be created in order to logically monitor different database performance areas. Each monitor can be independently started and stopped. The group operations, such as starting and stopping of several monitors at once for the same or multiple Informix instances, are also supported.

To create a monitor, enter a logical name, choose the database instance that you want to monitor, specify the monitor type and optionally provide a detailed description so that the purpose of this monitor could be easily understood by other DBAs. All user-defined monitor configurations are stored on the Sentinel server side and are accessed by Server Studio client when it connects to the Sentinel server.

If you need to apply similar monitors to several IDS instances, you can utilize the Replicate Monitor operation that allows you to use any existing monitor as a template for replicating a monitor definition for other database servers. After the 'Replicate monitor' operation is performed, the multiple instances of the template monitor are created using derivative names (by adding <_Number> to the original monitor name). You can edit each of these newly replicated monitors independently to adjust for differences in the database schemas or performance characteristics. For example, after replication of a QA server monitor to a production server, you might want to change the new monitor alert levels because production server might have a different configuration parameters and hardware.

Creating Monitors

To create a new monitor, select the New Monitor shortcut menu on the Monitors folder in the Sentinel – Automation Explorer. The New Monitor Wizard will take you through the steps necessary to create a new monitor.

Selecting Informix Performance Parameters

Sentinel server provides parameters for the following six categories of Informix performance metrics:

- **Server-level** parameters provide values for an entire Informix instance, such as READ CACHE %, USER CPU, SHARED MEMORY, CHECKPOINT_DURATION, % of used logical logs
- **DBSpace-level** parameters allows collecting data growth over time and setting alert notifications for conditions when the data exceeds specified thresholds. Alert notifications for a space-level parameter, provide a list of DBSpaces that exceeded the specified threshold value for this parameter.
- **Chunk-level** parameters allows measuring data growth over time as well as I/O activity on an individual chunk level. Alert notifications for a chunk-level parameter provide a list of chunk ids that exceeded the specified threshold value for this parameter.
- **Table/Index-level** parameters allows collecting data growth over time, I/O and locking activity and setting alert notifications on individual tables or indexes. For example, you can record BUFREADS parameter values for 3 tables and 5 indexes. Alert notifications for a table-level parameter provide a list of tables/indexes that exceeded the specified threshold value for this parameter.
- **Session-level** parameters allow collecting I/O, locking, transaction commit/rollback activity for users, client hosts or individual user sessions. Alert notifications for a session-level parameter provide the list of sessions/users/hosts that exceeded the specified threshold value for this parameter.
- **Virtual Processor-level** parameters allow collecting CPU and ready queue activity Informix virtual processor classes. Alert notifications for a VP-level parameter provide a list of VP classes that exceeded the specified threshold value for this parameter.

Selecting OS Performance Parameters

When an Informix instance is installed on UNIX or LINUX host, Sentinel server provides the functionality to monitor underlying OS host parameters.

Important: Sentinel uses Telnet or SSH access to underlying host workstation to retrieve OS-level performance parameters so that you must properly configure host access protocol and login parameters in the Server Studio Connection Manager.

Server Studio Connection Manager can be invoked using Tools → Connection Manager top-level menu and switching to Host Access tab.

Sentinel server provides parameters for the following five categories of OS performance metrics:

- **General**-these are OS host instance level parameters, such as free physical memory on the host, number of OS logins, paging activity or CPU usage for an entire host.
- **CPU**-level parameters allow monitoring activity on multi-processor systems. Parameters, such as percentage of time CPU is busy per each physical processor can be monitored per each installed processor.
- **Disk**-level parameters allow measuring I/O activity for OS disks. This category includes parameters, such as number of read and write operations per second, percentage of time disk was busy servicing requests, average disk service time, etc. Alert notifications for a disk-level parameter provide a list of OS disks that exceeded the specified threshold value for this parameter.
- **Volume**-level parameters allow monitoring the size and collecting the growth of the data on OS volumes over time.. Alert notifications for a volume-level parameter provide a list of volumes that exceeded the specified threshold value for this parameter.
- **Network**-level parameters allow monitoring and collecting network activity on OS host over time. This category includes parameters such as number of concurrent TCP/IP connections, number of received or sent TCP/IP packets, number of network packets errors.

For each parameter, you can define its own refresh interval on which the values are retrieved from the monitored Informix instance. For most of the parameters (excluding percentages and number of connected user sessions) monitors actually measure the delta between the current value of the parameter and its previous value. So that the value of a given parameter, which you see in the monitor, represents the growth of the activity which occurred during any given refresh interval.



Important: The absolute value of the parameter activity depends on the refresh interval. For example, assuming an equal server load over a period of time, 100 disk reads during 10 minute refresh interval will appear as a 600 hundred disk-reads during 1 hour refresh interval. Keep it in mind when you enter alert threshold values for these delta parameters. Every time a refresh interval is changed for the

parameter that has an associated alert threshold setup, that threshold value most probably become invalid and you will have to adjust it. A good practice for selecting threshold values is to run the monitor without any alerts for a period of time to learn what values are representative of the normal activity levels for your system (these values will vary from one system to another). Once the normal value for a particular parameter under a specified refresh interval is established, then you can set the appropriate thresholds for alert notifications. It does not apply to percentile and ratio parameters, which are not directly affected by changes in refresh intervals, such as CACHE READ% or COMMIT/ROLLBACK ratio.

If you want to apply the same refresh interval for multiple parameters, set the required value in the Default Refresh Rate field located at the bottom of the Parameters Selection panel. To save the collected values into historical database, select the Save checkbox on the right of the parameter name.

Defining Filters

Your monitor definition might include parameters from Table/Index, Space, Chunk, Session, Unix (CPU), Unix (Disk), Unix (Volume) categories, for example, DBSPACE FREE SPACE or TABLE BUFFREADS. In this case, a list of objects for each category has to be specified in order to limit the scope of Sentinel sever, such as saving data in a repository and checking alert conditions.

For example, you might have 2000 tables in your database, among which only 100 critical tables require continuous monitoring. Also you might have some tables in your database that always exceed an alert condition and you want to exclude them from alert checking process but still record their performance parameters into Historical Repository or display them in real-time. It also allows you to avoid an excessive Informix database load when retrieving unneeded monitoring information and helps prevent overloading of the historical repository with unnecessary data.

Just to give an idea of the required Historical Repository space, here is a sample calculation of the required disk space per hour if you choose to monitor and save table-level parameters every 1 minute for 2000 tables:

- 2000 (tables) * 350 (bytes) * 60 (minutes) = 42,000,000 (in bytes, or ~40 Mb).

To define a filter for a particular parameter category, switch to this category and press the Add button. Sessions category allows to specify filters for one or more user session ID, user login name or client host workstation.

Defining Alerts

New Monitor Wizard contains facilities to define alerts that can issue notifications when the value of a monitored parameter exceeds a user-defined threshold level and can trigger an autonomic corrective response by executing a predefined custom job. These jobs are user defined and can be comprised of user-defined OS scripts, OS commands, SQL scripts or stored procedure calls. For information on how to define alerts, see "Creating User-Defined Alerts" section of this guide.

Defining Real-Time Graphs

Real time graphing for monitor parameters can be defined when there is a need to visually chart the values as they are being retrieved by the Sentinel server. The real-time graphs are displayed when a monitor is running and opened in the Server Studio UI. Defining real-time graphs is optional for monitor operations. Moreover, real-time graphs can be dynamically added or removed from a running monitor.

Use real-time graphs to chart parameter values in real-time and visually inspect the affect of SQL executions on the performance of your system. For example, configure a session level monitor for your SQL Editor session, execute SQL statements from the selected SQL Editor and immediately see the impact of these statements on a set of different parameters that are being charted in real-time. This allows avoiding populating the repository with unneeded historical data.

For Server-level parameters, specify the parameter which you want to chart, choose the graph color and the chart area name where it should be displayed. Each real time graph is displayed in Chart Area which is a single XY graph panel. Multiple parameters can be charted in the same area to see correlation between selected parameters. For example, you can choose to display Disk Reads and Disk Writes graphs in the same chart area and Informix CPU Used and UNIX CPU Used in another chart area. Each graph in a chart area has its own value axis. You can move graphs to a different chart area at any time.

For Table/Index, Space, Chunk and Session parameters, in addition to graph color and chart area, a list of database objects has to be specified. For example, you can set up a filter to include 20 tables in the Table/Index category in order to collect historical data but graph only three most critical tables in real-time.

The real-time chart areas are organized into separate tabbed panels located at the bottom of the monitor panel. These tabs are called display areas. Each display area can contain multiple chart areas. Each chart area can be moved to a different display area at any time. To create a new display area, just type a new name in the Display Area column in the grid when adding new real-time graph. Display areas are automatically removed when all charts are moved out of them.

To add a real-time graph:

1. Choose the parameter name from the dropdown list in the grid
2. If the selected parameter belongs to one of these categories: Table, Space, Chunk, Session, Unix (CPU), Unix (Disk), Unix (Volume), the Object selection dialog box will be opened. Choose required objects and press the OK button. Multiple objects can be selected in this dialog and will be added simultaneously to the real-time definition grid.
3. Choose the color for each real-time graph. If you want to define a custom color, choose More... from the dropdown menu.
4. Choose a Chart Area for each graph. You can place multiple graphs in the same chart area. Default chart area name is Chart1. To rename it, just type

a new chart name in the provided field.

5. Optionally choose the value range for the value axis of the graph. Default option for the value range is <Auto>. When this option is selected, value axis is automatically scaled from 0 to maximum value in the chart. If you want to specify a particular value range, choose <Custom> from the dropdown and enter a minimum and maximum range values.

Editing Monitors Configuration

You can edit all attributes of a monitor, such as list of parameters, data collecting settings, alert definitions, object filters and real-time graphs at any time. If a monitor is running, the modifications will take effect immediately – you do not have to restart the monitor.

- To modify the monitor definition, you have to open it first. You should be the owner (creator) of the monitor to be able to edit it.
- To open the monitor, select it under the Monitors folder in the Sentinel – Automation Explorer and choose Open shortcut menu.
- To add, delete and modify monitor parameters, open the monitor and press Edit Parameters toolbar button.
- To modify object filters, such as list of tables or user sessions, open the monitor and press the Filters toolbar button.
- To modify the alert conditions, open the monitor and press the Alerts toolbar button.

Replicating Monitors

If you have similar Informix instances, you can replicate similar monitors created and tested for one of the instances. To replicate an existing monitor, choose the Replicate item from the shortcut menu of the selected monitor. After monitor structure is replicated for another Informix instance, it can be independently edited and adjusted for this particular instance. For example, an alert threshold level might need to be adjusted or the list of user names in session level monitoring might have to be different from the list of the original Informix instance. After replication, monitor names are created by adding “_#” suffix to a template monitor name, where # is a sequential number. You can rename these monitors afterwards.

Deleting Monitors

To delete a single monitor, select it under the Monitors folder in the Sentinel – Automation Explorer tree and choose the Delete from the shortcut menu.

To delete multiple monitors, open the Properties panel for the Monitors folder using the Monitors folder’s Properties shortcut menu. Select required monitors in the grid using CTRL+Click or SHIFT+Click operations. Delete the selected monitors using the Delete shortcut menu on the grid.

Note You cannot delete running monitors. You have to stop them first using the Stop shortcut menu.

Running Monitors

To start a single monitor, select the monitor under the Monitors folder in the Sentinel – Automation Explorer and choose the Start from the shortcut menu.

To start multiple monitors, select the Monitors folder and choose Properties from the shortcut menu. The list of monitors will be displayed in the Properties Panel. Using CTRL+Click or SHIFT+Click operations select multiple non-running monitors in the

grid and choose the Start from the grid shortcut menu.

Creating New Sentinel Connection

After the first installation, Server Studio is configured to connect to the local Sentinel server installed on the same workstation where Server Studio is installed. This single Sentinel server connection is displayed in the Sentinel-Automation tab as 'Sentinel Server (localhost)'. You can start working with this instance of the Sentinel server from Server Studio environment without having to perform additional setup or configuration steps.

In order to connect to remote Sentinel server instances, you have to explicitly add Sentinel connections to the Sentinel-Automation explorer. For example, you might have a dedicated Sentinel server for each installation of your Informix in this case you would add a new Sentinel connection in the Sentinel-Automation explorer in order to use Server Studio client to automatically connect and work with all of these Sentinel instances. To install Sentinel server on remote workstation, use the advanced mode of Server Studio and Sentinel server setup program. It can run in console mode and does not require graphical OS shell. Open readme.html file which is installed in <AGS> directory to read more on the installation options. Once the remote Sentinel is installed and started, you can create a connection to it from Server Studio using the following steps:

1. In Sentinel – Automation tab, select Sentinel Connections folder.
2. Choose New Sentinel from the shortcut menu
3. Provide a logical name for this Sentinel connection under which it will be displayed in the explorer tree. Enter the host name or IP address on which Sentinel server is running and the port number (default is 10556).
Optionally, clear the Auto-Connect checkbox if you do not want to automatically establish connection to this Sentinel server when Server Studio application starts. You will be able to use the explicit Connect operation on the Sentinel node to manually establish a connection.

Note Sentinel server has to be running on a remote host before you can create a connection to it.

ALERTS

Alerts provide a mechanism for sending email, SMS or Server Studio notifications when certain events happen either on the Sentinel or Informix side. These events are divided into Sentinel system alerts, user-defined Informix performance and Sentinel job alerts. Use the Alert Viewer to view and delete generated alert events. To open the Alert Viewer, select the Alert Events node under the Sentinel-Automation and choose the Open item from the shortcut menu.

User-Defined Alerts

The user-defined alerts are created as a part of the monitor definition. User-defined alerts can be configured to be received as Server Studio, email or mobile text message notifications. These alerts are triggered when the actual monitored parameter values exceed the specified thresholds. In order to avoid false positives or short term peaks, you can specify how the number of time a value has to exceed the threshold before the alerts sent out. To avoid getting too many alerts, an option is provided to block alerts if it was already triggered during a specified time period. All external alert notifications such as emails or mobile text messages are delivered via SMTP protocol, using SMTP server definition provided by the user as part of the monitor definition. The specified SMTP server should be accessible from the host on which Sentinel server is running. The difference between email, pager and mobile-phone messages is in the amount of text characters in the message (unlimited for an email message and up to 256 characters for a mobile-phone text message).

Each user-defined alert has its own severity, a list of notification methods and an optional action or a job that should be automatically executed when this alert event is triggered. The notification methods include Server Studio popup notifications and email, pager or mobile phone messages. Server Studio notifications always place an exclamation mark indicator on the corresponding monitor node in the Server Studio. You can also specify whether a popup dialog box should be displayed in Server Studio when a particular alert event occurs. A custom text message can be defined to contain short instructions on what should be done when a particular event occurs.

Multiple alerts can be created for the same parameter to differentiate between different severity levels. For example, you can create a warning alert when a dbspace is 80% full and a critical alert when a dbspace is 95% full. When an alert event is generated, it always includes the information about the alert's severity level, database instance where the alert event occurred, parameter value that exceeded the defined threshold, the actual threshold level and additional information about objects related to the alert event, such as list of tables, space names, user session ids, chunks ids, etc.

You create alert definitions at the time when you create a new monitor or by editing an existing monitor definition. Alert definitions are constrained by the list of parameters and the set of the database objects selected in the monitor.

Creating User-Defined Alerts

The alert mechanism allows you to issue notifications when the value of a monitored parameter exceeds user-defined threshold level. You can define conditions to alert you about practically any operational condition of your Informix server that might require attention (i.e., dbspace is almost full or a large number of deadlocks, etc.). Each alert condition has:

- a user-defined severity level
- parameter threshold value
- method of notification, such as Server Studio popup, email, pager or mobile phone
- an optional command to be executed on the monitored Informix server to automatically correct the problem or collect additional information about the operating environment

You can define multiple alert conditions for each monitor parameter with different actions to take when different threshold levels are exceeded. For example, you can configure to send visual alert with the severity level Warning when dbspace is 80% filled and send email and pager notification with a severity level Critical and automatically execute script that adds a chunk when dbspace is 99% filled.

To create an alert:

1. Press the Add button to add the new row in the Alerts grid.
2. From the dropdown list in the grid, choose the monitor parameter for which you want to define an alert condition.
3. Choose a conditional operation, such as > or <, and enter the threshold value.
4. Choose the severity level from the dropdown in the grid.
5. In the Alert Action panel located at the right of the Alerts grid, select the Popup Message checkbox if you want to see the popup dialog box in the Server Studio when this alert event occurs. Otherwise the alert event will be displayed in the Alert Viewer panel and also indicated by the exclamation mark on the monitor node under Monitors folder in the Sentinel – Automation Explorer.
6. Choose the value for the Notification Delay parameter. This parameter determines the period of time before server notifies about the same event that happens with the same object. For example, when a BUFFREADS parameter for a particular table exceeds an alert threshold for the first time, the notification will be sent out immediately. After that if the value for this parameter is still high during the next checking cycle but the Notification Delay is set to 1 hour, the alert event will not be sent out for another hour. . If you choose <NO DELAY> option, you will be notified every time the value for the table exceeds the threshold level. For example, if the table has a long period of high activity, you will receive 3 notifications during one

-
- hour period with a parameter running at a 20 minute refresh rate interval.
 7. Choose the value for the Notify After Event Occurs (n) Time(s). When this parameter is set to 1 (default), you will be notified every time the monitored value exceeds the specified threshold. Set this parameter to a higher number, if you want to be notified only when the same alert event happens several times in a row. For example, if this value is set to 2 for TABLE-BUFFREADS parameters and the threshold condition is "TABLE-BUFFREADS > 5000000" at 10 minutes refresh interval, you will be notified only when the value of the TABLE-BUFFREADS for some table stays greater than 5000000 during two sequential refresh intervals (20 minutes). It allows to avoid being notified when short term high peak events occur and to bring attention only to situations when the high activity condition persists over time.
 8. Choose the email, pager or mobile-phone notification method for this alert, if required. You can specify multiple emails, phones and pagers to send a message to in case of this event.
 9. Enter the user-message which will be part of this alert event. This message, for example, might contain instructions on how to resolve the issue associated with this alert event.
 10. Optionally choose a job, which should be executed on the target Informix host when this event occurs. The job can be a user-defined script, OS command, SQL script or a stored procedure call. The job should be created and tested using the Jobs folder before you can use it in the alert definition panel. See Jobs help topic for additional information on job creation, testing and execution.

For "OS Command" jobs, you can provide optional command arguments that will be appended to the job's command call when an alert is triggered. This mechanism provides you with the ability to provide alert-specific parameters for your script. For example, if you have an external logging utility that takes the text of the message as a command line argument, you can enter the following text in the Job Arguments field:

- "read cache < 80%"

The argument text is appended to the command line as is – therefore, you should provide all proper quote characters to separate text arguments containing spaces.

Alerts which are defined for table/index, chunk, dbspace or session level parameters require that a list of applicable database objects be specified in the Filters page of the Monitor Wizard.

Jobs Alerts

Job failure alerts are automatically generated when a user-defined job, such as a backup or UPDATE STATISTICS fails. This type of an alert event is considered to be a system wide and will be sent out to all Sentinel users. If the Sentinel administrator email is properly configured in the Sentinel Connection Configuration

Options tab, this alert is also emailed. An exclamation mark is displayed in the Sentinel – Automation Explorer tree for the job node that triggers an alert.

The job failure alert message contains a database instance name as well as the job name and an error text. You can see a job execution details by pressing the Details button in the Alert Viewer. If the failed job is triggered as a result of a user-defined monitor alert, the user-defined alert message also displays an exclamation mark to bring to your attention that an associated job failed.

Schema Audit Jobs Alerts

Schema Audit Job's alerts are generated when schema audit job detects a difference between a reference schema and target database schema. The alert details area provides information on which database schemas were compared and how many new differences were found. To see the detailed report that shows all found differences in database object's SQL syntax, press Show Report button in the Details panel of the Alert. You can also access generated Schema Audit reports in Schema Audit Reports folder under Change Management folder of the Sentinel server node. See Change Management section of this guide for more details on Schema Audit Jobs, Alerts and Reports.

NOTE: When a schema audit job runs continuously using a specified schedule, it generates an alert only the first time when a particular change is detected. After that a new alert will be generated only when additional changes are detected which were not included in the first alert. For example, if a schema audit job runs every 6 hours and a column was added to a database table, the alert event with this particular change report will be generated only 1 time during this day – not 4 times. However, each schema audit report always shows all changes between a reference schema and a database target schema – not only new changes that triggered the particular alert event.

System Alerts

System alerts are triggered under following situations:

- Monitored database instance goes off-line. The Sentinel server will automatically keep attempting to recover a failed database connection. All affected monitors automatically resume execution when a database connection is recovered. Another system information alert a connection is recovered.
- The Sentinel server is shutting down. All active monitors and scheduled jobs at this point are being gracefully stopped. . All affected scheduled jobs and monitors will automatically resume after the Sentinel server is re-started.
- Some of the monitored Informix sessions have been closed on the Informix server side and are not available for monitoring any more. The affected monitor continues execution after this event is sent.
- The internal Sentinel application error.

Viewing Alert Events

To view generated alert events you use the Alert Viewer panel. There are multiple ways to open the Alert Viewer:

1. In the Sentinel – Automation Explorer, select the Alert Events node and choose Properties from the shortcut menu.
2. From any existing monitor node in the Sentinel – Automation Explorer, choose the Alert Viewer from the shortcut menu.

The Alert Viewer panel consists of two areas: the alert event list and alert details. The alert event lists shows all alert events produced by the Sentinel server. You can filter it by an Informix instance or the number of last events. Sorting by the DB Instance, Severity and Date/Time of event can be performed by clicking on the grid headers. The details area shows the details for an event selected in the events master list. The left most column in the alert grid displays an exclamation mark when there was a failure executing a job associated with an alert or when the Sentinel server failed to send an email notification for the alert event.

JOBS

A Sentinel Job is either a SQL or OS shell script which can be executed on a schedule or triggered by an alert. It can perform a variety of administrative tasks, such as a database server backup, UPDATE STATISTICS operation or collection of OS environment information when database server performance degrades below a specified threshold level. You can easily define most of your standard existing server administration scripts as Server Studio jobs in order to get the following advantages:

- You can create a single graphical console for all your scripts for all instances of managed database servers. With this console you do not have to switch to multiple telnet terminal sessions and memorize different physical script names and directories on different hosts under different operating systems. You have a single centralized list, under which each job has clear logical name, description, last execution time, status and history. It displays the next execution time as well, if the job is scheduled for an execution.
- You can simultaneously start multiple jobs on multiple servers. For example, you can start UPDATE STATISTICS jobs for all of your servers at ones. It does not matter which operating system these servers run on.
- You can see all failed jobs in one single screen instead of checking the status of multiple terminal windows or multiple log files. You can also configure the system to send you email, pager or mobile phone alert message when critical job fails.
- You can schedule job executions using centralized, highly flexible scheduler utility. The scheduler works equally well for all servers, regardless of their operating systems, thus eliminating the need for different, operating system-dependent scheduling utilities.
- You can execute a job as a response to any user-defined monitor event. For example, to execute a flash checkpoint command when the physical log is almost full.
- Entire job execution history for all jobs with all text outputs is logged. So if you use a job to collect statistical information about your system, all data is logged automatically in the repository database, without you having to write special utility for it.

And best of all, in order to get all these advantages, a minimum effort is required to define a job. This includes entering logical job name, specifying an existing script host and location and defining failure or success text pattern string.

The Sentinel server executes a job when:

- The user explicitly requests to execute it from the Server Studio UI
- Job is scheduled to be executed using the Job Scheduler
- Job execution is triggered by a monitor alert event

A Sentinel job can be defined to be executed against any host workstation with which Sentinel server can communicate via Telnet, SSH or JDBC protocol. However, in majority of cases you will define jobs to be executed against the same host where your database server resides so that you could automate database

maintenance tasks for that database server.

The Sentinel server executes jobs in the background, thus there is no need to keep Server Studio opened when a long job is started. For example, you can start a 3 hour backup job from the Server Studio, make sure that the execution is started, and then close the Server Studio application and logout from your workstation. The next morning you can start the Server Studio and check on the status of the job as well as to see the text output produced by the job during the execution.

When job executions fail or is cancelled by a user, a system wide alert is generated and sent out immediately. You can view job failure alerts using the Alert Viewer. The complete log of job execution history can be reviewed in the Job History tab of the Job Properties panel.

To view the text output produced by the job while the job is being executed, use the Last Execution tab in the Job Properties panel. This is useful when executing long running OS commands that might produce some progress messages, such as Informix ONTAPE utility, and you want to monitor the job progress. There is no need to keep Server Studio in order to keep receiving job progress updates. Any time Server Studio is reopened, all current status updates are automatically sent out by the Sentinel server. Status updates, for a multi statement SQL script, contain the number of the statement which is currently being executed on the Sentinel server side.

Configuring Jobs

There are multiple ways to create a new job:

1. Select Jobs folder in the Sentinel – Automation Explorer→<Sentinel Server>→Performance Management and choose New Job from the shortcut menu.
2. Use File → New → New Job top-level menu.
3. Use Tools → Monitoring → New Job menu.

Once the Job Editor is open, enter a logical name and choose the job type. The name has to be unique as it will identify this job in the Sentinel-Automation Explorer tree and in the alert response selection lists. The two job types which are available in the dropdown selection will be discussed in the section below.

Defining “OS Command”-level Jobs

The OS Command Job allows you to execute operating systems command or shell script via Telnet or SSH protocol. When executing a job, the Sentinel server automatically establishes a connection and logs in to the host specified in the job and then executes a specified command or script using one of the above mentioned protocols. The text output produced by the command execution is captured and stored in the job history log in the Sentinel's repository. In addition, the output is sent in real-time to the connected Server Studio client that requested this job execution.



Important: When defining the OS command, you have to make sure that it does not require any interaction with the user – i.e. prompting for any input parameters. Once launched, the command should be executed to the end. If the command, such as Informix ONTAPE utility, needs some input, you should develop a wrapper shell script that calls the command and feeds it with all the required input. For example, you can redirect a command's input stream to a text file and put all required responses to the command's prompts into this file. In addition, you should always provide a reasonable timeout value, if you know that there is a chance that the command might prompt for some input. When the time-out is provided, Sentinel automatically breaks the command execution after the specified time period expires and the command is not finished yet. For example, if you know that your backup job usually does not take longer than 1 hour, you should specify 2 hours time out interval. When the command times out, the job failure alert is automatically sent to the user who launched the job's execution.

A text pattern which identifies the status of the job execution can be supplied to Sentinel server to automate recognition of OS shell command failure. Usually each command provides some type of text output at the end, which identifies either successful completion or some type of an error. By default, if no status text patterns are provided, Sentinel considers a job completed successfully if it did not time out and returned to the operating system prompt. For example, ONTAPE utility displays text 'Program over' when it terminates successfully. If your command does not provide a standard text message for either success or failure situation, you can develop a wrap shell script, that analyzes the status of the command execution and provides some standard text output pattern that can be used when defining OS Command Job. The status text patterns are case-sensitive.



Tip: Always test a defined job interactively by executing it in Server Studio and viewing the produced status and text output before using it as a response to a monitor alert. To test a defined job, select it under the Jobs node in the Sentinel – Automation Explorer and choose the Execute from the shortcut menu. To test SSH/Telnet connectivity, choose the Edit and press on the Test connection button. This operation does not actually execute the job command; it only tests if the required connection can be established with the specified host and port name.

Defining SQL Script/Stored Procedure Jobs

You can define a job as SQL script. The script can consist of any number of valid SQL statements separated by a semicolon (;) character. It can also contain stored procedure calls. An example of a SQL script job would be a set of UPDATE STATISTICS commands which can be generated in Server Studio.



Tip: If you do not have Telnet or SSH access for your system, you can use a stored procedure to execute an operating system command on the server host..

SYSTEM Statements

The SPL SYSTEM statement can be used from a stored procedure to execute an OS command, as the following syntax examples show:

- SYSTEM “OS command”; or
- SYSTEM variable_name

If the supplied expression is not a character expression, an *expression* is converted to a character expression before the operating-system command is made. The complete character expression is passed to the operating system and executes as an operating-system command. The operating-system command that the SYSTEM statement specifies cannot run in the background. The database server waits for the operating system to complete execution of the command before it continues to the next procedure statement. The procedure cannot use a value or values that the command returns.

If the operating-system command fails (that is, if the operating system returns a nonzero status for the command), an exception is raised that contains the returned operating-system status as the ISAM error code and an appropriate SQL error code. In DBA- and owner-privileged procedures that contain SYSTEM statements, the operating-system command runs with the permissions of the user who is executing the procedure.

Specifying Environment Variables in SYSTEM Statements

When the operating-system command that SYSTEM statement specifies is executed, no guarantee exists that the environment variables that the user application set are passed to the operating system. To ensure that the environment variables that the application set are carried forward to the operating system, enter a SYSTEM command that sets the environment variables before you enter the SYSTEM command that causes the operating-system command to execute.

Examples of the SYSTEM Statement: UNIX

The following example shows a SYSTEM statement in a stored procedure. The SYSTEM statement in this procedure causes the UNIX operating system to send a mail message to the system administrator.

```
CREATE PROCEDURE sensitive_update()
.
.
.
LET mailcall = 'mail headhoncho < alert';
-- code that evaluates if operator tries to execute a
-- certain command, then sends email to system
-- administrator
SYSTEM mailcall;
.
.
.
END PROCEDURE; -- sensitive_update
```

You can use a double-pipe symbol (||) to concatenate expressions with a SYSTEM statement, as the following example shows:

```
CREATE PROCEDURE sensitive_update2()
DEFINE user1 char(15);
DEFINE user2 char(15);
LET user1 = 'joe';
LET user2 = 'mary';
```

```

.
.
.
-- code that evaluates if operator tries to execute a
-- certain command, then sends email to system
-- administrator
SYSTEM 'mail -s violation' ||user1 || ' ' || user2
|| '< violation_file';
.
.
.
END PROCEDURE; --sensitive_update2

```

Examples of the SYSTEM Statement: Windows NT

The following example shows a SYSTEM statement in a stored procedure. The first SYSTEM statement in this procedure causes the Windows NT operating system to send an error message to a temporary file and to put the message in a system log that is sorted alphabetically. The second SYSTEM statement in the procedure causes the operating system to delete the temporary file.

```

CREATE PROCEDURE test_proc()
.
.
.
SYSTEM 'type errormess101 > %tmp%tmpfile.txt |
sort >> %SystemRoot%systemlog.txt';
SYSTEM 'del %tmp%tmpfile.txt';
.
.
.
END PROCEDURE; --test_proc

```

The expressions that follow the SYSTEM statements in this example contain two variables, **%tmp%** and **%SystemRoot%**. Both of these variables are defined by the Windows NT operating system.

Using Jobs

A Sentinel job can be executed manually, by scheduler or triggered by a monitor alert. When the same job is triggered simultaneously from several sources, it is executed only once. For example, if the backup job was triggered by an alert and then subsequently is triggered by the scheduler process, the invocation from the scheduler will be ignored by Sentinel.

Manual Job Execution

To start job execution from the Server Studio, select a job in the Jobs folder of the Sentinel – Automation Explorer and choose the Execute from the shortcut menu. You will see the progress of the execution in the Properties panel of this job. The job is executed by the Sentinel server in the background so that you can close Server Studio at any time – it will not cancel the job execution process. However, stopping of the Sentinel service will cancel a running job. To explicitly cancel a running job, select this job in the Jobs folder and choose the Kill from the shortcut menu. It might take longer to cancel the SQL Script job, because it requires interruption of the process on the database server.

To view the full execution history log, select the Properties from the job shortcut

menu and switch to the History tab located at the bottom of the Job Properties panel. The Last Execution tab contains information about the status of the last job execution.

Scheduled Job Execution

You can schedule a job execution using the Schedule tab of the Job Editor dialog box.. To open the Schedule tab, choose the Schedule item from the job node shortcut menu. The scheduler has the following options:

- Enable Schedule checkbox – select this box to enable the execution of the job by scheduler. Clear this box if you want to stop scheduling executions of the job by the scheduler. Clearing this check box does not delete the selected schedule but simply disables it. You still can re-use the same scheduler options later when you select Enable Schedule checkbox again.

Job Execution in Response to Alert Event

To execute a job in response to a monitor alert event:

1. Select the monitor in the Sentinel – Automation Explorer→<Sentinel Server>→Performance Management→Monitors.
2. Choose the Open from the shortcut menu. The monitor document will be opened.
3. In the monitor document press the Alerts button or choose Monitor → Alerts top-level menu. The Alerts editor will be opened.
4. Select the required alert condition in the Alerts grid, select the Execute Job checkbox in the Alert Actions panel
5. Choose a previously created and tested job from the Jobs dropdown list.



Tip: Always run the job from the Server Studio to make sure that it runs properly before assigning it as an alert action.

Editing Jobs

To edit the job definition or rename the job, select it in the Jobs folder of the Sentinel – Automation Explorer and choose the Edit from the shortcut menu. Once the Job Editor dialog opens, make the desired modifications in the job parameters, including the scheduler parameters. To apply the changes press OK button.

Deleting Jobs

To delete a single job, select it in the Sentinel-Automation→<Sentinel Server>→Performance Management → Jobs folder and choose the Delete from the shortcut menu. To delete multiple jobs, select the Sentinel-Automation→<Sentinel Server>→Performance Management → Jobs folder and choose Properties from the shortcut menu to open the job list in the Properties panel. Select multiple jobs in the grid using CTRL+Click or SHIFT+Click operations and choose the Delete from the shortcut menu to delete all selected jobs. When a job is deleted, it is also deleted from all monitors that reference this job. You will be prompted to confirm job

deletion if there are monitors that use this job.

SQL CAPTURE

The SQL Capture process allows you to periodically retrieve SQL statements that are being executed on the Informix server. You can explicitly specify which SQL statements have to be retrieved based on session user login names, client host, Informix session ID, or SQL statement execution statistical parameters, such as estimated cost, estimated number of retrieved rows or number of sequential scans. You can also filter retrieved SQL statements by the database objects that they refer to or by statement type such as INSERT, UPDATE, DELETE, EXECUTE, DDL. Retrieved SQL statements can be displayed on the screen in real time and/or recorded in the Sentinel repository database for further analysis. You can analyze captured SQL statements using historical SQL Viewer tool.

Creating SQL Capture Process

To create a new SQL Capture process, expand Sentinel server node in the Sentinel – Automation Explorer→<Sentinel Server>→Performance Management, select SQL Capture folder and choose New from the shortcut menu. The New SQL Capture Process wizard will open. In the first page of the wizard, you can choose the following parameters:

- **DB Instance** – the IDS server, from which you want to capture SQL statements.
- **Save in the Repository** – select this box if want to save captured SQL statements in the repository.
- **Refresh Rate** – choose the required SQL retrieval interval. It is important to understand that not all SQL statements executed on the server can be captured. If the statement is executed faster than the specified refresh rate and the client process closes the open SQL statement cursor or immediately executes another SQL statement, than the first statement will not be captured. The purpose of the SQL Capture process is to highlight poorly performing SQL statements – not all SQL statements.

Filter options allow you to specify the user sessions and performance parameters for the SQL statement that you want to capture.

- **Filter by User** – use this radio button if you want to retrieve only the SQL statements executed by specified users. Press the Add button to choose user login names.
- **Filter by Host** – use this radio button if you want to retrieve only the SQL statements executed from specified client host workstations. Press the Add button to choose client host workstations.
- **Filter by Session ID** – use this radio button if you want to retrieve only the SQL statements executed by specified client sessions. Press the Add button to choose client session IDs.
- **Performance Parameters** – you can use this grid to enter threshold values for SQL statements performance parameters, such as estimated cost or estimated

number of affected rows. It allows you to intercept only problematic, poorly performing statements out of the entire universe of SQL statements executed on the database server.

- *Filter by Database Object* – press the Add button to add a list of database objects which should be referenced in the captured SQL statements. For example, if you want to capture only statements that insert, update or select from 'order' and 'product' tables, add these two tables to the filter list.
- *Filter by Statement Type*– select SQL statement types of SQL statements that you want to capture. For example, if you do not want to capture only DML statements, select INSERT, UPDATE and DELETE options,

The second page of the wizard provides the following fields:

- Name – give a logical name to this SQL Capture process. It will be displayed under this name in the SQL Capture folder of the Sentinel – Automation Explorer.
- Description – optionally enter a descriptive text that explains the purpose of the SQL Process. Start after creation – select this box if you want to launch this SQL Capture process immediately after it is created on the Sentinel server. Alternatively, you can use the Start Process shortcut menu or button on the SQL Capture process panel to launch it at any time.



Important:

When a SQL Capture is defined to capture SQL statements for real-time display purpose only (option to collect the data in the repository was not selected), the process is automatically stopped when the viewer is closed. When the option to collect data is selected in the SQL Capture process, it will keep running even when the viewer is closed.

Viewing Captured SQL Statements

To see the retrieved SQL statements in real-time, use the Open shortcut menu on the running SQL Capture process in the Sentinel – Automation Explorer→<Sentinel Server>→Performance Management→SQL Capture tree. A separate panel for this SQL Capture process will open. It has a SQL Capture area that displays captured SQL statement in real-time. The controls in that area are the same as for the SQL Viewer Tool described below.

To see SQL statements recorded in the Sentinel repository by SQL Capture processes, use History SQL Viewer Tool. You can easily pinpoint the worst performing statements and see who and how often executes them by using the SQL Viewer Tool. You can also capture a sequence of SQL statements for a particular session and use them to build meaningful load test scenario that replicates behavior of this session in the Server Studio Benchmark Runner.

The SQL Viewer Tool allows you to specify the time interval, for which you want to review the collected statements. It also allows you to sort displayed SQL statements chronologically, or based on the estimated performance statistics. You can filter displayed statements by session ID, if you are interested in reviewing SQL executed by particular user thread chronologically.

There are multiple ways to open SQL Viewer Tool:

- 1) Choose History → SQL Viewer shortcut menu on Sentinel server node in the Sentinel – Automation Explorer.
- 2) Choose History → SQL Viewer shortcut menu on any SQL Capture process node in the Sentinel – Automation Explorer

The SQL Viewer Tool panel consists of the selection wizard and results panel. In the selection wizard, you can specify the time interval for analysis, list of session IDs that have to be displayed and optional filters by database object and SQL statement type.

The SQL Viewer Tool results panel consists of the following components:

- A grid that displays all SQL statements within the specified time interval for the selected session IDs.
- SQL statement details area that displays information about the statement selected in the grid.
- Sort By dropdown list that allows you to sort data in the grid.
- Filter By Session ID dropdown list that allows you to display only the statements for the selected user session.

PERFORMANCE HISTORY REPOSITORY

The Historical Data Repository is used by the Sentinel server to store the following information:

- Parametric historical performance data collected by all monitors.
- SQL statements collected by SQL Capture processes.
- Alert events generated by all monitors.
- Job execution history.

The Historical Data Repository consists of several tables that can be queried using SQL statements as well as using special analytical graphical tools provided in the Server Studio. Sentinel contains an embedded SQL database that serves as a data repository. This embedded repository is used by default. You can also configure Sentinel to use an external Informix database as a data repository.

Viewing Historical Performance Statistics

You can view the collected historical data using graphical analytical and query tools provided in the Server Studio. Server Studio graphical tools allow viewing of the collected data as a timeseries or hourly, daily or monthly histogram. You can also export time series data into a text file for further analysis using external analytical tools, such as Microsoft Excel.

Graphical Tool

There are multiple ways to open a new historical data graphical viewer:

- From a monitor node, choose the History → Graphs shortcut menu. It will open a selection screen pre-populated with performance parameters that exist in the selected monitor definition.
- Choose the target Informix connection node in the Object explorer and select the Monitors → History Graphs shortcut menu.
- From the Performance Management node menu, choose Monitor → History → Graphs.

Graphical Viewer Selection dialog box consists of a grid in which you can specify Informix performance parameters that you want to see.

It includes the following fields:

Dataset name – logical name for the selected historical performance data. The name in the field will be the default file name for this set if you decide to save it.

Instance - Informix server instance name. The dropdown contains all IDS instance names for which there is data available in the Sentinel historical data repository.

Parameter—select an Informix performance parameter to display in a historical

report. The dropdown field contains a list of parameters for the Informix instance selected in the Instance column. Select parameters from different categories if you would like to see correlation between them.

Data Collection Interval - this dropdown list contains continuous time intervals on which data for a selected parameter were collected by Sentinel monitors. For each continuous time interval displays the data retrieval rate at which data was collected (for example, every 30 seconds, every 5 minutes, etc.) If you want to define a custom time interval that does not exist in the dropdown list, choose the Custom item to open a custom interval dialog box.. This dialog box lets you enter the start and end value for new custom interval. The newly created custom interval is added to the dropdown list and you can re-use it for any parameter in the grid – not only for the parameter for which this custom interval is created.

Object – select a specific database object for which data was collected in the repository. This field is not required for IDS Server, Unix General, Unix Network level categories of the parameters.

Chart Area - graphs for selected parameters are displayed in chart areas. You can place multiple parameters in the same chart area in order to see correlation between parameters. Parameters that are placed in the same chart area share a time axis but have a dedicated value axis. Value range can be customized for each parameter in the chart area. The default value range is <Auto> that means that the value axis is automatically scaled to display the highest value. You can also choose to place the selected parameters into different chart areas. To define a new chart area, just type the name of the new area in the provided field. This name will serve as a header for a chart. For example, parameters such as user_cpu and system_cpu can be put into one chart area titled 'Informix CPU Usage' while parameters such as read, writes into 'Informix I/O Activity'.

Color – select the color of the parameter graph in the specified chart area.

Graph Type – you can display historical data using four graph types: time-series, daily histogram, weekly histogram, monthly histogram. Histogram charts provide an easy way to identify peak Informix activity during a day, week or month.

Change Interval button – allows you to change the time interval for selected parameters in the grid. This is an easy way to re-use previous graph definitions for the new time interval (e.g. current week or month)

Replicate button – allows re-use of the selected graph definitions for a different Informix instance and/or time interval.

Save to File button – allows saving all graph definitions into a file for later re-use.

Open From File button – allows opening graph definitions previously saved in this dialog box.

Once you made all required selections, press the OK button to start data retrieval. Graphical Viewer Document opens and displays multiple chart areas with the required data.

In the graphical viewer panel, use the Display As shortcut menu on the selected graph area to switch between time-series, hourly, weekly and monthly histograms. Use the Export shortcut menu on the selected graph to export the graph time-series data into a text file.

Use the Properties shortcut menu, to open a Selector dialog box again. If you only modify the display properties, such as colors, chat area names or move graphs from one chart area to another one, the currently retrieved data is simply reused after you apply changes. However, if you add parameters or change time periods, data for the affected parameters will be refreshed from the repository which might take a few second.

To zoom in into a portion of the chart, highlight the area by holding the left mouse key down.

There are several options available to save the graphs or the underlying data. To save the underlying data in a CSV (Comma Separated Values) format to export the data for other analytical tools, press on the Export data button. To save the graph definition, press on the Save Report Definitions button. To save the generated charts into image files, press the Save as Image Files.

Query Tool

The Query Tool allows you to directly query the data stored in the Historical Repository tables using SQL statements. The retrieved data is displayed in the grid and can be also exported to a text file for further analysis by external analytical tools. There are multiple ways to open a historical data graphical viewer:

- From a monitor node, choose the History → Queries shortcut menu.
- From a monitor's top-level menu, choose the Monitor → History → Queries.
- From a top-level Sentinel menu, choose the Query Repository Data menu.

The Query Tool panel contains an editor area, result set area and the repository structure tree. In the editor area, you can type a single or multiple SQL queries, separated by semicolon “;” character. Use the SQL menu or the shortcut menu to execute all or highlighted statements in the editor and export generated result sets to text files. You can save queries created in the editor by using the File → Save menu. To open a previously saved query, choose the SQL → File Open menu after the Query Tool panel is opened. To execute a query, use the SQL → Execute menu or Execute toolbar button. To export a retrieved data set, use the SQL → Export All Data menu or shortcut menu on the selected result set grid.

Repository Structure

The Repository Structure tree in the Historical Data Query Tool is provided as a convenient in-place reference for the structure of the repository. Each table represents a single category of parameters and a column in the table holds data for a specific performance parameter. Expand table nodes to see the list of all columns in each table. Select the Sentinel → > Query Repository Data top-level menu to open the Query Tool.

LOAD TESTING WITH BENCHMARK RUNNER™

Benchmark Runner™ allows you to create custom load test scenarios that simulate behavior of your database application, execute them from hundreds of virtual user sessions and measure client-side response time as well as impact of the application on the database server resources and overall performance.

You can also use the Benchmark Runner™ to analyze performance of your database applications before and after you change the database server configuration settings. For example, after certain changes in the server configuration, the performance of OLTP applications can improve but performance of the decision support applications can decrease. Benchmark Runner scenario allows you to create a consistent performance benchmark test for all kinds of transactions and run this test after each configuration modification to see the performance impact.

The Benchmark Runner™ load test scenario consists of multiple user-defined tasks. Each task is an SQL script that can include one or multiple SQL statements. Each task can be executed from one or multiple simultaneous virtual user sessions, using random or fixed interval between executions. Benchmark Runner™ calculates average, minimum and maximum execution time for each task, permitting you to profile and analyze response times based on the number of user sessions performing the same operations.

One more way of measuring the performance impact of different Benchmark Runner tasks is to use Sentinel monitoring facility. It allows observing in real-time how such metrics as Informix CPU, memory and disk I/O, lock consumption, etc. are affected by the execution of the specific SQL statements.

For example, to create a realistic test, create several tasks in the Benchmark Runner that will represent a new order entry, reporting and order review operations (or any such tasks that resembles the behavior of your application). Increase the number of user session to match the average load that you expect for each type of an operation. Each task can be executed either at a fixed time interval (each task is executed every N number of seconds) or random interval (each task can be executed at random during N number of seconds). After running this load test scenario for a period of time and monitoring the server parameters, using user-defined monitors described in the Monitors section of this guide, you will see how well your application performs under real-life stress conditions. By increasing the number of sessions and frequency of the task execution you can also validate how well your system is designed for concurrent data access by measuring number of dead locks, lock waits and transaction rollbacks.

Creating Benchmark Runner™ Scenarios

To create a new Benchmark Runner™ load test scenario:

1. Choose the File → New → Benchmark Scenario menu to open a new load test scenario panel.

-
2. Select a database instance for the load test scenario from the Connection dropdown list box or create a new connection by pressing ‘...’ button.
 3. Press the New Task button in the load test scenario panel to create a new task.
 4. Use the Task Name field to modify default name for the task. You can give it a logical name, such as Sales Order Entry or Accounting Report.
 5. Use the SQL Script editor to create a script for the task. Multiple SQL statements should be separated using ‘;’ character. Optionally, you can import an existing SQL script from a text file by pressing the Import button.
 6. If your script requires parametric values for any of the SQL statements, press the Edit Variables button to define load test scenario variables against an existing text data files as explained in the next section.
 7. Use the Number of Sessions field to enter the required number of the virtual user sessions for this load test scenario task. When the load test scenario is executed, Benchmark Runner™ will create the number of physical independent database connections you specified and each of these connections will execute the SQL script defined in the load test scenario task.
 8. Use the ‘Interval Between executions’ field to specify a delay between sequential script executions within one virtual user session. You can specify either random or fixed intervals. For example, if you specify 600 sec random interval, it means that the next script execution in the same virtual user connection might happen anytime between 0 sec and 600 seconds. Enter zero if you do not want to have any delay between script executions.

Repeat above steps to define additional load test tasks for the Benchmark Runner™ scenario.. After you’ve completed defining your entire load test scenario, save it in a file by pressing the Save button. To start the load test scenario execution, press the Start button.

Using Variables in Benchmark Runner™ Scenarios

You can create parametric SQL statements in your Benchmark Runner™ load test scenarios by substituting hard-coded values in any type of an SQL statement (such as: SELECT, INSERT or UPDATE) with values dynamically retrieved from external text data files. You can define data retrieval rules (Read Order) for the load test scenario variables to simulate any application logic.

For example, to simulate a real-life load on an ACCOUNT table where different users query the table and then update it, you need to use a parametric query with a random retrieval method. This forces Informix server to perform actual I/O operations and not just buffer a specific data set as the SELECT statements hit random physical chunks. This type of a scenario will give you better work load simulation as opposed to running SQL statements with hard coded or sequential

primary key values.

To define a load test scenario variable:

1. In the task definition screen, press the Edit Variables button. The Edit Script Variable dialog box will open.
2. Press the New Variable button
3. Choose a logical name for the variable, for example: "ID_Order"
4. Choose the Read Order option. If you select the Random option, the values will be read from the text data file randomly. If you select the Sequential option, the values will be read sequentially and restarted from the beginning when the end of the file is reached.
5. Select the external data file that contains the test values. It should be a text file with one column of data, separated by an end of the line or tab character. For string values, do not use quotes. For datetime and interval values, use any supported string representation that can be converted to a corresponding datatype.
6. To add additional variables, repeat from step 2.

Use \$variable_name syntax to reference the variable anywhere in the SQL script. For example you can use them in a WHERE clause of a SELECT statement or in a VALUES clause of an INSERT statement.

Example,

```
SELECT id_payment, amount, discount
FROM tbl_payments
WHERE id_order = $ID_Order
```

Editing Benchmark Runner™ Scenarios

To edit a Benchmark Runner™ load test scenario:

- 1) Choose File → Open → Benchmark Scenario menu.
- 2) Select the saved scenario file and press Open button
- 3) Edit the scenario parameters in the Benchmark Runner™ panel and press the Save button to save the changes.

CHANGE MANAGEMENT

Change Management Option streamlines database configuration change management by simplifying the tedious and error-prone task of maintaining traceable records of the database schema states changes and automatically notifying administrators when a change from a baseline schema is detected for an active database. Using a scheduled schema audit job, you can compare periodically a single 'baseline' database schema, for example, a snapshot of a tested QA database, against multiple production databases that can reside on different Informix servers. This job will detect any modifications made in a production schema, saving you a lot of time and efforts when you need to keep multiple production server instances synchronized with a single source database schema. Change Management subsystem allows maintaining traceable records of database security states, such as user's and role's permissions and track security changes for audit purposes. It also enables capturing periodic database schema version control snapshots ("VCS") and retaining them in the built-in repository for easy roll-out, roll-back and recovery of lost or damaged database objects.

Change Management Repository

Sentinel server hosts Change Management Repository that stores schema version snapshots created for Informix databases as well as schema audit reports. Change Management Repository maintains a catalog of database object attributes similar to Informix internal catalogs providing large flexibility in comparing different versions of schemas, generating schema synchronization scripts, showing object dependencies and object security permissions.

You can choose one of the following repository types:

- Build-in Cloudspace database (default option) – the default location of embedded Cloudspace repository database is
 <AGS>\userdata\sentinel\stores\vcs
 where <AGS> is a Sentinel server installation directory
- External Informix database – the Informix database name is
 'sentinel_cmo'

To choose the Change Management Repository type, select Sentinel server node, select the Configure Sentinel from the shortcut menu and switch to Repositories tab.

Schema Audit Jobs

Schema Audit jobs allow monitoring for schema changes on multiple Informix servers and issue email notifications when differences are found. Email alerts can optionally contain an HTML difference report along with a synchronization script which allows to rollback the changes. Schema audit jobs can automatically take schema snapshots of the monitored Informix servers for record keeping or for manual reviewing.

Schema audit jobs provide you with the following advantages:

- Schema audit jobs can be scheduled to run on the specific time or run periodically, acting as a schema audit tool.
- Using a single job, you can choose to compare multiple databases or compare the same source database schema against multiple target databases located on multiple Informix instances.
- Schema comparison reports can be automatically emailed to you or to several DBAs when differences are found. These reports include detailed difference analysis in HTML format as well as pre-generated synchronization script required to rollback all or a subset of changes.

In addition, schema audit jobs can create a version snapshot of the database state at the time when differences are found so that you can review them after the changes in the live database are rolled back.

Creating Schema Audit Job

To open New Schema Audit Job wizard, switch to Sentinel – Automation tab, expand Sentinel server node, expand Change Management folder, right-mouse click on Schema Audit Jobs folder and select New Schema Audit Job from the shortcut menu. On the first page of the wizard, enter a job name that describes a purpose of the job. Choose the 'source' database schema that will serve as a baseline for a comparison job. It can be a 'live' database schema, as for example, your QA server database schema, or a Server Studio schema snapshot previously created and stored in the Sentinel repository. If you want to create a new snapshot at this point, choose Version Snapshot radio button and press Create New Snapshot button. See Version Snapshots section of this guide for additional information on version snapshots functionality. You can include one or more databases into a single schema audit job. For example, you can create a job that compares all databases on your source Informix server instance to all databases on one or multiple target server instances. This model works well when you have multiple deployment targets for your Informix applications and all targets should have exactly the same schema as the source target. Press the Next button to define target Informix instances. Select one or more target instances and press the Next button. The third page of the wizard attempts to automatically match the selected databases or snapshots for the source Informix instance to databases with the same name on selected target instances. If your target database or databases have different names than a source database name, you can manually specify a target database name for each source database in the Target Database column in the grid. Press the Next button to define schema comparison options. You can choose to compare only selected group of objects, ignore physical storage properties, such as dbspace names, include or exclude user permissions comparison, etc. Press the Next button to define how you want to be notified when

the differences in schema are found. You can choose to send an email message that can optionally include a schema difference report and a pre-generated synchronization SQL script. Press the Next button to define a schedule for a newly created job. If you do not define a schedule right away, you still will be able to execute schema audit jobs manually or add the schedule at later time after the job is created. Press the Next button to define actions that should be taken when the schema audit job fails during execution. Optionally, define actions to be taken when job execution succeeds. For example, you might want to send an email notification when a scheduled schema audit job fails because it cannot establish a connection to a target Informix server. Schema audit job can also execute another job in case of failure and/or success. After the schema audit job is created on Sentinel server, you can access it in the Sentinel-Automation tab under Sentinel Server → Change Management → Schema Audit Jobs folder. If the job was scheduled, it will be executed as per specified schedule. Otherwise, you can execute it manually by selecting the job in the tree and choosing Execute from the mouse popup menu. To edit the job, choose the Edit from the mouse popup menu. All schema difference reports created by a particular schema audit job are available in the job's Properties panel. Alternatively, all difference reports created by all schema audit jobs executed on Sentinel server are available in Sentinel Server → Change Management → Schema Audit Reports folder under Sentinel – Automation tab.

Editing Schema Audit Job

To open Schema Audit Job Editor, switch to Sentinel – Automation tab, expand Sentinel server node, expand Change Management folder, expand Schema Audit Jobs folder, select required job and choose Edit from a popup menu.

Schema Audit Job Editor dialog box consists of five tabs:

Job Definition – edit the job name and a list of database schemas to be compared. Press Edit button located under the grid to invoke the schema objects selector dialog box. You can change a list of schemas currently defined for the job as well as add a new source/target pairs or delete existing pairs.

Schedule –modify a schedule for the job. If you want to temporarily disable a schedule, uncheck Enable Schedule checkbox located at the top of this tab. You can execute the job manually at any time using Execute popup menu at job's node in the Object Explorer tree. The job can be found under Sentinel's server node in Change Management → Schema Audit Jobs folder.

Comparison Options – choose which categories of database objects should be compared as part of this schema audit job. For example, you can choose to include or ignore user privileges, such as GRANT and REVOKE permissions to database objects.

Successful Execution Actions –define which actions to take when the schema audit job is executed successfully and schema differences are found.

There are several actions available:

Send email message with attached detailed difference report and pre-generated synchronization script to specified administrators

Create a snapshot of entire target database schema in Sentinel's Change Management Repository so that it can be analyzed at later time

Execute user-defined job, previously created as Sentinel job (OS script or SQL stored procedure)

Failure Actions – define actions to take when the schema audit job fails for any reason. For example, it can fail when one of target database servers goes off-line or a network connection from Sentinel server workstation to target Informix servers becomes unavailable. You can choose to send notification email to a list of administrators or execute a user-defined job.

Deleting Schema Audit Job

To delete Schema Audit Job, switch to Sentinel – Automation tab, expand Sentinel server node, expand Change Management folder, expand Schema Audit Jobs folder, select required job and choose Delete from a popup menu.

Viewing Schema Audit Reports

To view Schema Audit reports created by all schema audit jobs, switch to the Sentinel – Automation tab, expand Sentinel server node, expand Change Management folder, and select Schema Audit Reports folder. Properties panel contains a grid with all reports generated by all schema audit jobs. To see the report details, select a report in the grid and press the Show Details button or double-click on the report in the grid. Each report details panel contains a list of all database objects that were modified, deleted or added and specific SQL syntax differences in each modified object.

To delete selected schema audit reports, select the required reports in the grid and press the Delete button. Once the Delete dialog box opens, press the OK button to confirm the delete.

If you want to delete all reports in a specified date range or reports created by a single schema audit job, press the Delete button without selecting any reports in the grid. You will be able to enter the additional information in the Delete Schema Audit Reports dialog box.

Version Snapshots

Version Snapshots function of Change Management subsystem allows performing the following tasks:

- Capture periodic database schema version control snapshots ("VCS") and retain them in the built-in repository.
- Create state-in-time snapshots of individual objects, such as: stored procedures, tables, views, etc.

-
- Maintain traceable record of database security states, such as user's and role's permissions and track security changes for audit purposes.
 - Preview versions of database schema objects using graphical tools and recover selected objects' schema
 - Use robust differential analysis facilities to detect changes from a baseline, between any two snapshots; or between the state of any currently active database and a given system snapshots stored in the in the built-in repository.
 - Easy roll-out, roll-back and recover lost or damaged database objects.
 - Version Snapshots can serve as 'baseline' schema for Schema Audit Job

To use Version Snapshots, you need to start Sentinel server on a workstation that will serve as your Version Control server and to which all developers and DBAs have access.

Creating Version Snapshots

You can create snapshots for:

- Single database
- Single database object, such as table or stored procedure
- Group of databases

Creating snapshot for a single database

Select a database in the Object Explorer tree and choose Version Snapshots → Create New Snapshot from the popup menu. You will be prompted to enter a label and an optional description for a snapshot. Enter a label – this is a text that will identify your snapshot in the list of snapshots – for example, QA2. Optionally enter a description text that provides additional information about a snapshot content. Press the OK button to initiate snapshot creation on the Sentinel server. If it takes a long to time to create a snapshot, press the Background button to close the dialog box. You will receive an alert from the Sentinel server when a snapshot is created.

Creating a snapshot for individual database object

Select a database object, such as a stored procedure or table, in the Object Explorer tree and choose Version Snapshots → Create New Snapshot from popup menu. You will be prompted to enter a label and optional description for a snapshot. Enter a label – this is a text that will identify your snapshot in the list of snapshots – for example, QA2. Optionally enter a description text that provides additional information about the content of the new snapshot. Press the OK button to initiate a snapshot creation on the Sentinel server.

Creating a snapshot for a group of databases

Select Version Snapshots folder under Informix server node in the Object Explorer tree and choose Create New Snapshot from popup menu. You will be prompted to select the list of databases for which you want to create a snapshot and enter a label and optional description for a snapshot. Enter a label – this is a text that will identify your snapshot in the list of snapshots – for example, QA2. Optionally enter a description text that provides additional information about the content of the new snapshot. Press the OK button to initiate snapshot creation on the Sentinel server. If it takes a long time to create a snapshot, press the Background button to close the dialog box. You will receive an alert from Sentinel server when snapshot is created.

Using Version Snapshots

To access a created snapshot for a database or an individual database object, select the corresponding node in the Object Explorer and choose Versions Snapshots → Show History from a popup menu. Snapshot History dialog box, which displays all available snapshots for a database or an object, will open. You can use this dialog box to view snapshot content, generate SQL script file for a snapshot, compare snapshot to a current object state or compare two selected snapshots.

Viewing Snapshot Content

To view snapshot content, select a snapshot and press the View button. Snapshot viewer will be opened in the Properties window. For a database snapshot, it will show a list of all objects in this snapshot. Select an individual object and choose Properties from the popup menu or double-click on the object to view its details. You can use Back button located on the main toolbar afterwards to return to the list of objects in a snapshot.

Generating SQL script based on a snapshot

To generate SQL schema script based on a snapshot, select a snapshot and press the Get button. Snapshot viewer will open in the new SQL Editor document.

Comparing Snapshot with a current database state

To compare a snapshot with the current database state, select a snapshot and the 'Current State' item in the grid and press the Diff button.

Comparing Snapshot with another snapshot

To compare a snapshot with another snapshot, select two snapshots in the grid and press the Diff button.

Deleting Version Snapshots

NOTE: you have to be Sentinel Administrator to delete version snapshots from the

repository.

There are two ways to delete version snapshots.

Deleting a group of selected version snapshots

To delete one or several snapshots for a database instance, switch to the Sentinel – Automation tab, expand the Sentinel server node, expand the Change Management folder, select Version Snapshots folder, select one or several snapshots that you want to delete in the Properties panel grid and choose the Delete from the popup menu. If you want to sort the list of snapshots based on a particular parameter, click on the required column header in the grid.

NOTE: to select multiple snapshots in the Properties grid, use Ctrl+Click mouse operation.

Deleting all snapshots for Informix server instance

To delete all snapshots for selected Informix instances, switch to Sentinel – Automation tab, select Sentinel server node and choose Maintenance → Purge Change Management Repository from the popup menu. Select all Informix instances for which you want to delete snapshots from the repository and press the OK button.

APPENDIX A: ADDITIONAL INFORMATION

APPENDIX B: