****

**Specification**

**for**

Modified by Vicky Harp

**3/5/10**

IderaLogo

Proprietary and Confidential to BBS Technologies, Inc.

**© 2010 BBS Technologies, Inc.; all rights reserved.**

# Revision history:

|  |  |  |
| --- | --- | --- |
| **When?** | **Who?** | **What?** |
| 2/22/2010 | Kurt | First Draft |
| 3/5/2010 | Kurt | Final |
| 3/22/2010 | Vicky | Updated to add history data |
| 5/12/2010 | Vicky | Added Appendix 1 |

# Table of Contents

Wait Stat Monitoring of Queries i

Kurt Goolsbee Modified by Vicky Harp i

1. Revision history: ii

2. Table of Contents iii

3. Requirements 4

3.1. Overview/Purpose 4

3.1.1. Related Customer Requests 4

3.2. Target Users 4

3.3. Feature/Function Market Requirements 4

3.3.1. Required Functions 4

3.4. FAQ 4

3.5. Open Issues 4

4. 5

5. Functional Design 5

5.1. User Interfaces 5

5.2. Installation and Upgrade 8

5.3. Permissions and other Required Configuration 8

5.4. Licensing Issues 8

5.5. Dependencies 8

6. Internal Design 8

6.1. Architecture 8

6.1.1. Collection Service 8

6.1.2. Management Service 8

6.1.3. Desktop Client 8

6.1.4. Repository 9

6.2. Installation Issues 9

6.3. Schedule 9

6.3.1. Work Breakdown and Sizing 9

6.3.2. Areas of Risk 9

7. Quality Assurance Considerations 9

7.1.1. Overview 9

7.1.2. Developer-Created Unit Tests 9

8. Documentation Consideration 10

# Requirements

## Overview/Purpose

An analysis of Waits indicates where SQL Server is spending lots of time waiting. By correlating the areas of the server that are waiting with other statistics it is possible to help narrow down possible causes of the waits. For instance, lots of time waiting on disk resources could indicate that you have a disk problem. Several competitive products allow the user to correlate queries with waits. The idea is to focus on tuning the queries that wait the most.

Competitive SQL Server monitoring products have the ability to monitor SQL Server wait statistics for queries are Confio Ignite and Quest Performance Analyzer and Symantec I3.

### Related Customer Requests

Unable to find a PR for the request but I have a pre-sale request for a customer currently using Symantec I3 to add similar functionality to SQLdm.

## Target Users

This feature will benefit database administrators looking to tune their SQL Servers by reducing the time the server has to sit idle waiting for resources. It will help the user narrow down a list of queries to focus on.

## Feature/Function Market Requirements

### Required Functions

* Users should be able to view the top waiting queries by database.
* Users should be able to view the top waiting queries by application.
* Users should be able to view the top waiting queries by client name.
* Users should be able to view the top waiting queries by user.
* Users should be able to view the top waiting queries by session.
* Users should be able to view the top waiting queries by statement.
* Users should be able to filter results by database, application, client, user and session.

## FAQ

None.

## Open Issues

How are we going to show the SQL statement in the graphs? One competing product shows a hash code for each query and allows the user to provide a friendly name to be shown.

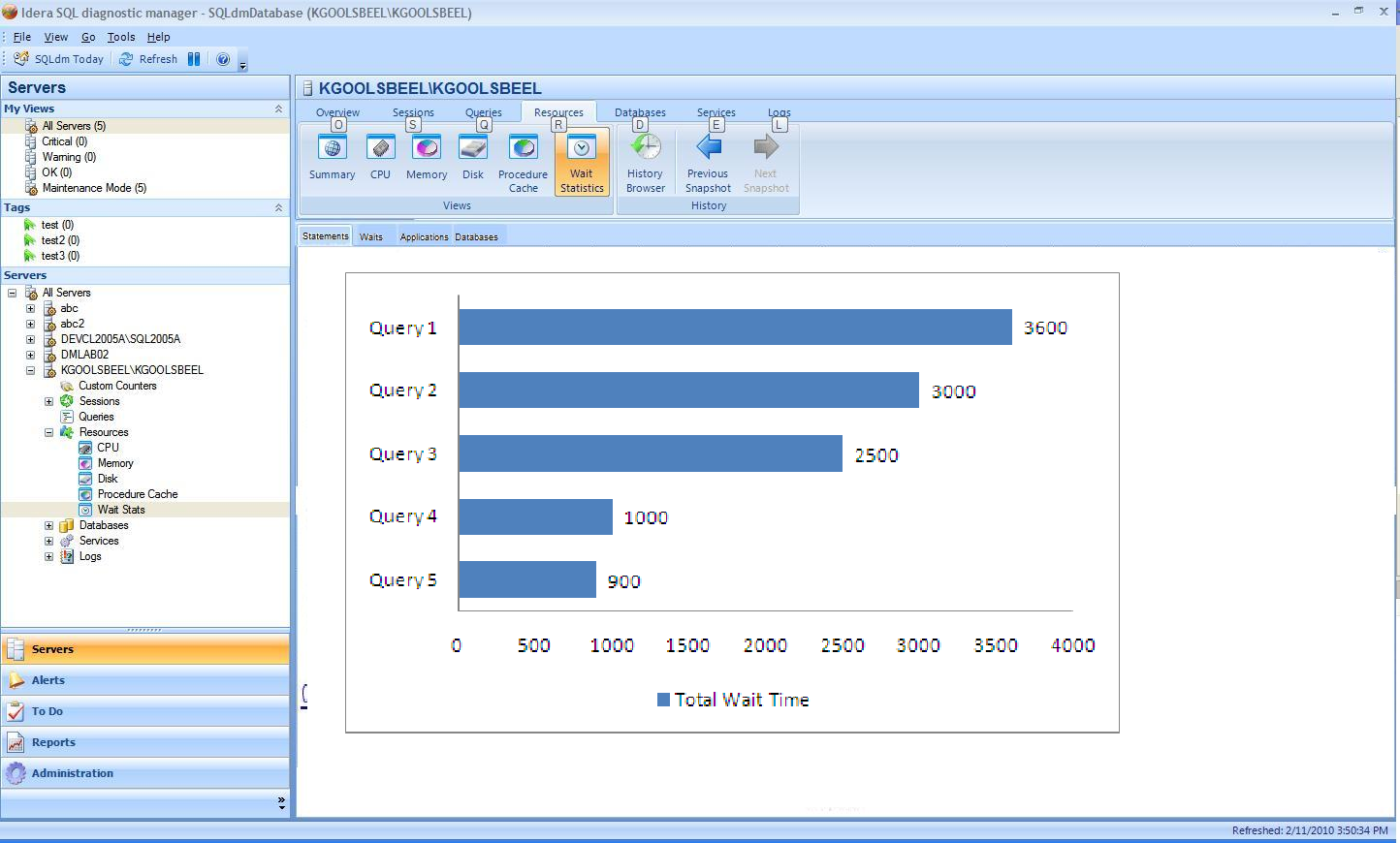
Answer: We will name the sql statements with a label plus a sequentially assigned number (Query 1, Query 2…). The hover text will show the query text.

# 

# Functional Design

## User Interfaces

This view will be available in real time, and also available in history browser mode within certain parameters discussed below. It is accessed using the Active Waits button on the Waits tab of the ribbon.



At the top of the view there will be a set of tabs that allow the user to navigate between the charts. The charts below don’t do a very good job of showing that they are stacked bars but there could be multiple chunks of colors shown for each bar.

All charts will be horizontal stacked bar charts with total wait time on the X axis.

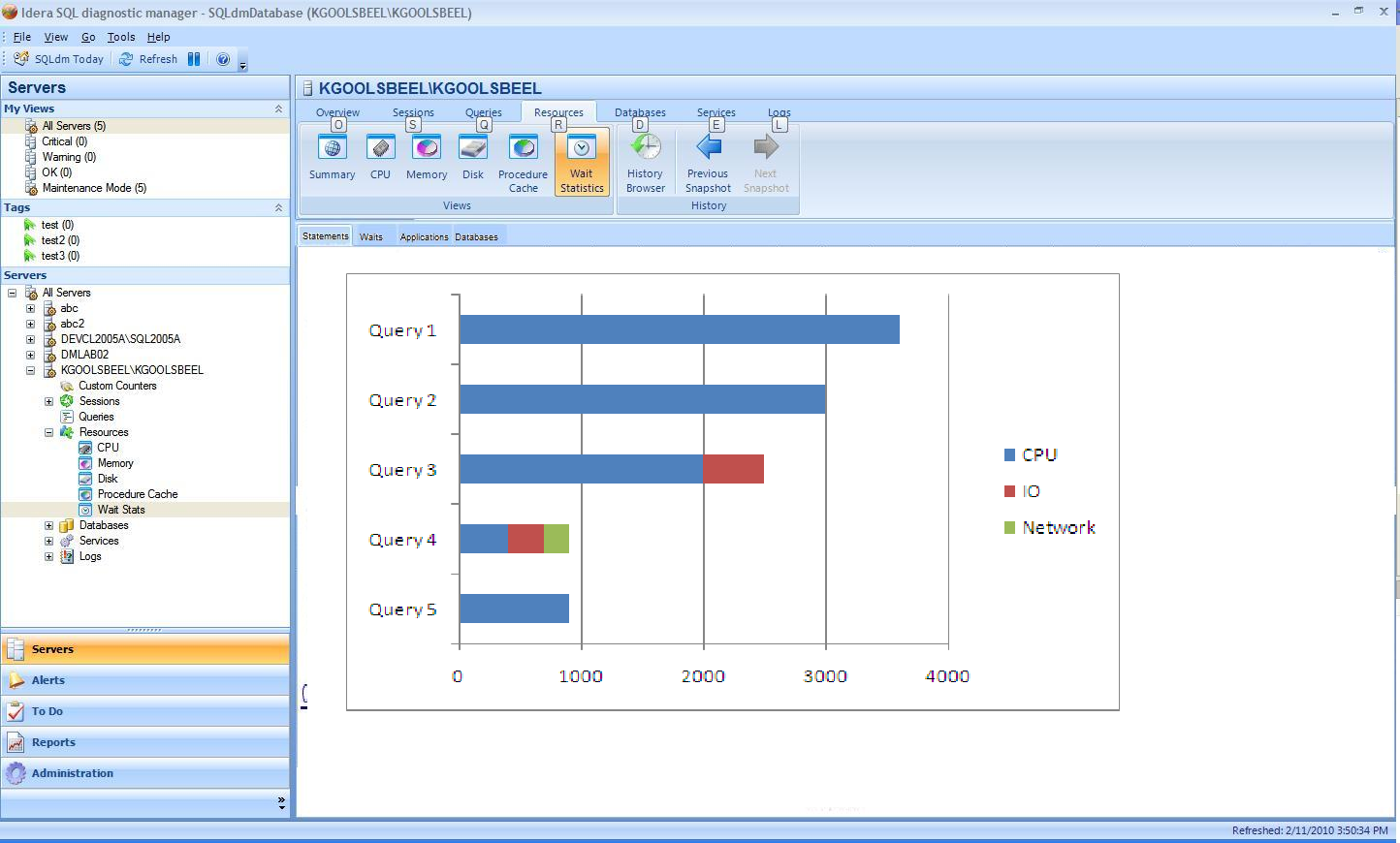
The series values are the wait times for each wait type.

Items will be sorted so that the longest wait times are at the top.

The legend will show wait type. Clicking on a wait type in the legend will show balloon help for the wait type.

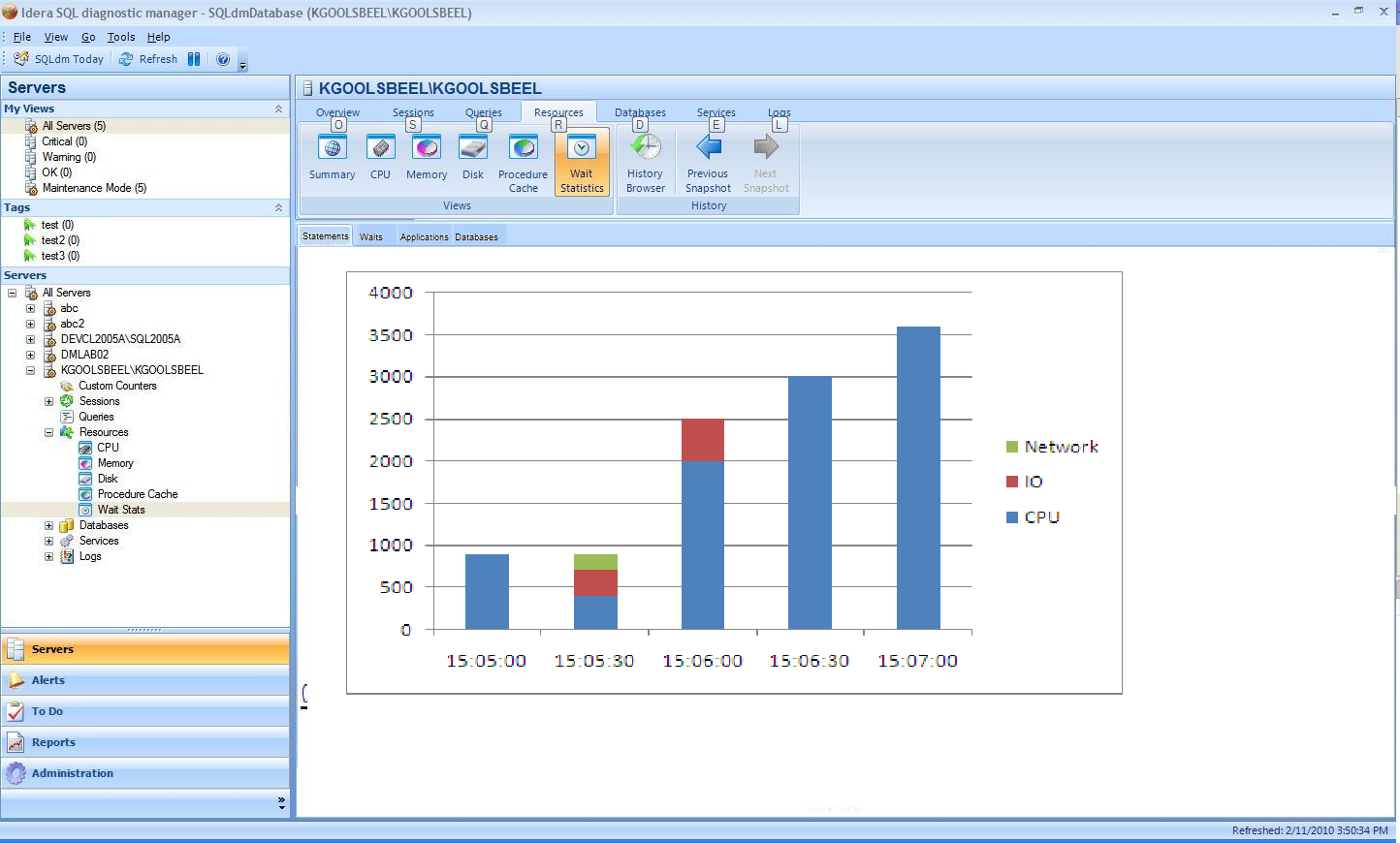
The tabs will be (also shown on the Y axis)

* Statements – Top X waiting SQL statements



* Waits – Top X Wait Types
* Applications – Top X waiting applications
* Databases – Top X waiting databases
* Machines – Top X waiting machines
* Sessions – Top X waiting sessions
* Users – Top X waiting users

We will also be able to show it by showing time along the x-axis…..



Each segment in the bar will have tooltip text that displays the wait time represented by that wait.

Each row on the Y axis will be selectable by clicking on the y axis label or the bar in the chart.

On the Statement chart the link will show you some basic execution statistics for the SQL statement. Selecting a Y axis link on any of the other tabs will add the selected value as a filter.

When a filter is added the user is navigated to the Statement view and the filtered tab is removed. The filter value will be added to a bread crumb trail above the graph.

The bread crumb will list the filters in the order they are added. The last crumb is not selectable. Selecting a prior crumb will delete all filters after the selected crumb and will navigate you to the Statement tab. Filters that are removed will cause the corresponding tab to become visible again.

Possible links to other product areas:

* Database – could provide option to switch to a database view for a selected database
* Query – could provide option to switch to the query view for a selected SQL statement
* SPID – could provide option switch to one of the session views for a selected session

Special handling of the Refresh button

The view is going to control its collection cycle independent of the normal foreground refresh. The refresh and pause buttons will behave like on/off switches. On entry the refresh button will be disabled. Clicking the pause button will suspend real-time monitoring and enable the refresh button. To resume monitoring the user will need to click the refresh button.

Data grooming

Chart data will be groomed using the console setting for real-time charts (default is 30 minutes). Thirty minutes may be too short for some people so we may need to add the ability to extend this for just the active waits view.

Special handling of historical data for this view

Because of the intense nature of per-query wait stats collection, historical wait stats will only be available if explicitly gathered by the user. It will be necessary to add a section to Server Properties for Wait Statistics Collection, allowing historical wait stats per query to be enabled for a time period up to 24 hours in length. The user should also be able to see when collection started if it is in progress, and the user should be able to cancel collection from this point. While a collection is in progress it should not be possible to extend the length of the collection – this is to prevent the timeframe from being continuously extended. These limitations may be removed once the feature proves to be stable over longer timelines. Presently we are also not planning to allow users to schedule this collection, though that may also be a change that is possible in the future. A link to the Server Properties dialog should be provided from the Wait Statistics by Queries view.

## Installation and Upgrade

No special considerations.

## Permissions and other Required Configuration

No special considerations.

## Licensing Issues

No special considerations.

## Dependencies

No special considerations.

# Internal Design

## Architecture

### Collection Service

Add a real-time probe to collect and summarize wait statistics gathered from the current session list. The probe needs to execute the query at least one time per second for a configurable period of time. The results of each query will need to be analyzed and summarized so that delta values for wait time are returned. This will require the Desktop Client to send the previous snapshot since we don’t retain this kind of stuff in the collection service for real-time probes.

The historical collection of wait statistics will be implemented as a separate scheduled collector from the scheduled refresh, which will run at between a 30 second and 1 minute interval and which will not be involved in alerting. While this collector is running, any desktop client which attempts a real-time collection should be attached to the running wait statistics history collector to prevent duplicate collectors.

### Management Service

The management service will need to pass through the real-time request to the collection service and return the results when completed.

### Desktop Client

A new view will be added to show the active wait statistics. The same data is used to drive all the charts. It just needs to be filtered and summarized to match the chart. Refer to section 5 for the details.

### Repository

The wait statistics by query will be saved to the repository for retrieval by the desktop client. While this area is not well defined yet, the data will probably be saved in a serialized fashion much like the current Session Details. The wait statistics history data will be groomed out with the rest of the server activity history data.

## Installation Issues

None

## Schedule

### Work Breakdown and Sizing

|  |  |  |
| --- | --- | --- |
| Component | Who | Sizing |
| Collection Service | Vicky | 10 days |
| Management Service | Vicky | 3 days |
| Repository | Vicky | 3 day |
| Desktop Client | Quentin | 14 days |
| Unit and Integration Testing | All | 2 days |
| **Total** |  | **32 days** |

### Areas of Risk

This is an entirely new form of collection for the collection service and should be considered high risk. This risk extends to almost every part of the collection service because a severe performance bottleneck could affect the performance of all on-demand and scheduled collections. Special attention should be paid to ensure that the wait stats collection starts and stops when expected and that wait stats collections do not stack up on each other.

# Quality Assurance Considerations

### Overview

As this is a high risk feature it will be delivered in multiple parts to allow the separate testing of the scheduled component and the on-demand component of the wait collection.

### Developer-Created Unit Tests

None expected

# Documentation Consideration

None

# Appendix 1 – Implementation Notes

A number of changes were introduced during implementation in response to QA and PM requests and in order to fulfill technical requirements. For completeness these changes are delineated below. The same appendix is being included for both the Server Waits and Query Waits specifications.

* The Server Waits View will have both a line chart and a stacked chart, as well as the ability to change between the chart types
* In Server Waits and Query Waits the legend will reflect Wait Categories and not specific wait types
* While it was discussed on and off, no pie chart will be included for the Server Waits view in this release
* In the Query Waits view two buttons will be added to the ribbon:
  + A link to the Server Properties page to allow the query wait statistics to be configured
  + A button to define the number of records returned in the charts – the Top X, defaulting to 10
* Where labels for a chart are too long and are truncated, the series should be right-clicked upon to view the full label
* In Query Waits, when switching to timeline mode the top X series being shown will be the series shown in the timeline mode as well
* This specification refers to drill-throughs or links to other product areas. This is not being implemented in this release.