



LANDESK Service Desk

Design and Configuration for a
Performant System

This document contains information, which is the confidential information and/or proprietary property of LANDESK Software, Inc. and its affiliates (referred to collectively as "LANDESK"), and may not be disclosed or copied without prior written consent of LANDESK.

To the maximum extent permitted under applicable law, LANDESK assumes no liability whatsoever, and disclaims any express or implied warranty, relating to the sale and/or use of LANDESK products including liability or warranties relating to fitness for a particular purpose, merchantability, or infringement of any patent, copyright or other intellectual property right, without limiting the rights under copyright.

LANDESK retains the right to make changes to this document or related product specifications and descriptions, at any time, without notice. LANDESK makes no warranty for the use of this document and assumes no responsibility for any errors that can appear in the document nor does it make a commitment to update the information contained herein. For the most current product information, please visit www.landesk.com.

Copyright © 2011, 2014, LANDESK Software, Inc. and its affiliates. All rights reserved. LANDESK and its logos are registered trademarks or trademarks of LANDESK Software, Inc. and its affiliates in the United States and/or other countries. Other brands and names may be claimed as the property of others.

Document number LDS\021 Issue 1.5

Contents

Contents	3
Design and configuration for a performant system	5
Introduction	6
Improving this document	6
Configuration	7
General configuration tips	8
Service Desk Console configuration advice	13
Design	17
General observation	18
Query design	18
Query configuration	19
Dashboard design	20
Process collections on Group	20
Avoiding large collections (unbound collections)	21
Configuring queries or filters below the window (tabbed queries)	21
After Read Calculations	22
Analysing a poorly performing system	23
General analysis steps	24
Checklist	28

Design and configuration for a performant system

You can find out about:

- *General configuration tips* on page 8
- *Service Desk Console configuration advice* on page 13
- *Design* on page 17
- *Analysing a poorly performing system* on page 23

Introduction

This document is a guide only for assisting with identifying possible performance improvements that can be applied to LANDESK Service Desk. Additional tips and checks that are unrelated to the application itself but which have been identified from experience of previous customer installations have also been included. There are many variables related to performance that are out of our visibility and beyond our control (for example network performance, resource and disk usage, infrastructure, etc). Usage is one of the most important variables affecting performance, and as such we can only provide advice in these areas which are not all-encompassing and are subject to change.

IMPORTANT: It is the customer's responsibility to validate these suggestions and to ensure that any changes made are tested before applying them to a live system.

We recommend that the customer defines what an acceptable performance level is as part of the implementation plan, and then within the test and acceptance plan validates that it is fit for purpose. Include and monitor an on-going maintenance plan to ensure that this is kept at an acceptable level with fine tuning as required.

LANDESK Service Desk is highly configurable and some designs may have a negative effect on performance. The purpose of this document is to help you to develop a system that balances the design requirements for the system with the performance requirements for the system.

This document assumes that you are familiar with the installation and basic architecture of the product – for more information, see the *LANDESK Service Desk Setup Guide*.

IMPORTANT: We recommend upgrading to the latest version of Service Desk as a pre requisite to troubleshooting performance issues.

Some observations made require knowledge of the underlying operating system and IIS (suitable for Windows administrators), database platform (suitable for a DBA) plus how the design of queries and windows are achieved within Service Desk (suitable for your Service Desk administrator).

There are three main sections to this document:

- **Configuration** – advice on how to set up and configure the system to maximise performance. This includes advice on database, IIS, and Console settings.
- **Design** – advice on how to use the Service Desk Console designers to design a system that maximises performance. The designers enable you to set up business objects, windows, processes, and queries in a wide variety of ways to maximise the possibilities for the final system. However, some design decisions can have a negative effect on performance – this document will highlight the design decisions that need to be used with care.
- **Identifying issues** – the final section offers advice on troubleshooting a poorly performing system, so that you can identify where to make changes to improve performance.

There are, of course, overlaps in each of these sections, so we recommend that you familiarise yourself with the entire document.

Improving this document

We are constantly looking to improve the information that we provide to help you to design and configure systems in such a way as to maximise performance. If you have any information that could be added to this document, or corrections and enhancements for this document, please e-mail DoctorDocs@landesk.com.

Configuration

This section contains tips on how to set up and configure the underlying system to maximise performance. This includes advice on database, IIS, and Console settings.

CAUTION: The most important piece of advice is to ensure that the environment where LANDESK Service Desk is being installed fulfils the requirements of the *LANDESK Service Desk Supported Platforms* document and the *LANDESK Service Desk Technical Specification & Architecture Guidelines*. Both of these documents are available from LANDESK.

In this section you can learn about:

- *General configuration tips* on page 8
- *Service Desk Console configuration advice* on page 13

General configuration tips

This section gives general advice on how to configure LANDESK Service Desk to optimise its performance. *Service Desk Console configuration advice* on page 13 gives information additional to this section, regarding optimising the performance of the console only. The advice in the *General configuration advice* section will also improve the performance of the console.

SQL Server: Deadlocks

If you are getting a lot of deadlock error messages on your system, first consider rebuilding indexes and updating statistics on the database (see *Database maintenance planning* on page 11).

If this does not help, and you are using Microsoft SQL Server, check whether or not Snapshot Isolation is enabled on your Service Desk database, and if it is not, consider enabling it.

NOTE: By default on databases created in 7.8 or later, this should be enabled, but on databases created in prior versions this might be disabled.

NOTE: For more information, see the following community article:
<http://community.landesk.com/support/docs/DOC-22128>.

NOTE: For more information about Snapshot Isolation, see Microsoft's documentation:
<http://msdn.microsoft.com/en-us/library/ms189050.aspx>

Process Workload business object

If you use the **Process Workload** business object to generate your Workload List, each time a process or process assignment is created or modified, or users or groups are modified within the Administration component, data is immediately entered into the database. If you do not use the Process Workload business object as the basis for your Workload list, you can disable this action, which will mean that processes are logged and updated more quickly. To do this, you modify the tps.config file.

Add the line:

```
<add key="PopulateProcessWorkload" value="False" />
```

...to **C:\ProgramData\LANDESK\ServiceDesk\servicedesk.Framework\tps.config**.

NOTE: See also the following article on the community: <http://community.landesk.com/support/docs/DOC-6463>.

Using multiple instances of Service Desk Framework

If you have a large number of users or a high volume of data being entered, creating more than one copy of the Service Desk Framework web application to point to your database could improve performance. This is because all database traffic (except for Web Access and reports that you run outside of Service Desk) goes through the Service Desk Framework. For information on how to create more than one copy of the Service Desk Framework using the LANDESK Configuration Centre, refer to the *LANDESK Service Desk Setup Guide*.

We recommend that you create an instance of the Service Desk Framework on the same server as your services (that is, on the Application Services server) and that the services (Mail Manager, Data Import, and so on) use this instance of the Framework. Instances of the Framework on the Web server will then be used exclusively by Service Desk users.

To see if creating multiple instances of the Framework improves performance, point just one of the Service Desk users (preferably one who uses the software heavily) to a copy of the Framework just for their use. If after a few days of monitoring they have seen an improvement, implementing multiple instances of the Framework could improve system performance significantly.

IIS Application Pooling on the Web Server

If you are sharing a web server with other web sites or applications, we recommend that you create a separate application pool for Service Desk called **LANDESK Applications** and move each Service Desk web application into this pool. We also recommend that you create a separate application pool for Web Access.

In addition you could further separate each of your LANDESK web applications into separate application pools and then ensure that each LANDESK application uses a different one. This will allow multiple dedicated worker processes and should help performance.

NOTE: For more information on IIS, see the Microsoft IIS website: <http://www.iis.net>

You can switch off the options to re-cycle application pools in IIS, or set these to recycle out of hours or during a period of reduced activity. Re-cycling an application pool is effectively the same as restarting the application, which then disconnects everyone and so reduces performance while the cache is refreshed. However, this can clear memory that has been taken up by the worker process.

The screenshot shows the 'Edit Application Pool Recycling Settings' dialog box. The title bar reads 'Edit Application Pool Recycling Settings'. The main content area is titled 'Recycling Conditions' and contains the following settings:

- Fixed Intervals:**
 - Regular time intervals (in minutes): [Text Box]
 - Fixed number of requests: [Text Box]
 - Specific time(s): [Text Box] Example: 20:00,00:00
- Memory Based Maximums:**
 - Virtual memory usage (in KB): [Text Box]
 - Private memory usage (in KB): [Text Box]

At the bottom of the dialog, there are four buttons: 'Previous', 'Next', 'Finish', and 'Cancel'.

LANDESK Application Server engine settings

You can configure the LANDESK Application Server engines to log different levels of event messages. Make sure that these are all set to log either Error or Critical. If they are set to Information or Trace for more than a short period of testing, the Application Log will quickly fill.

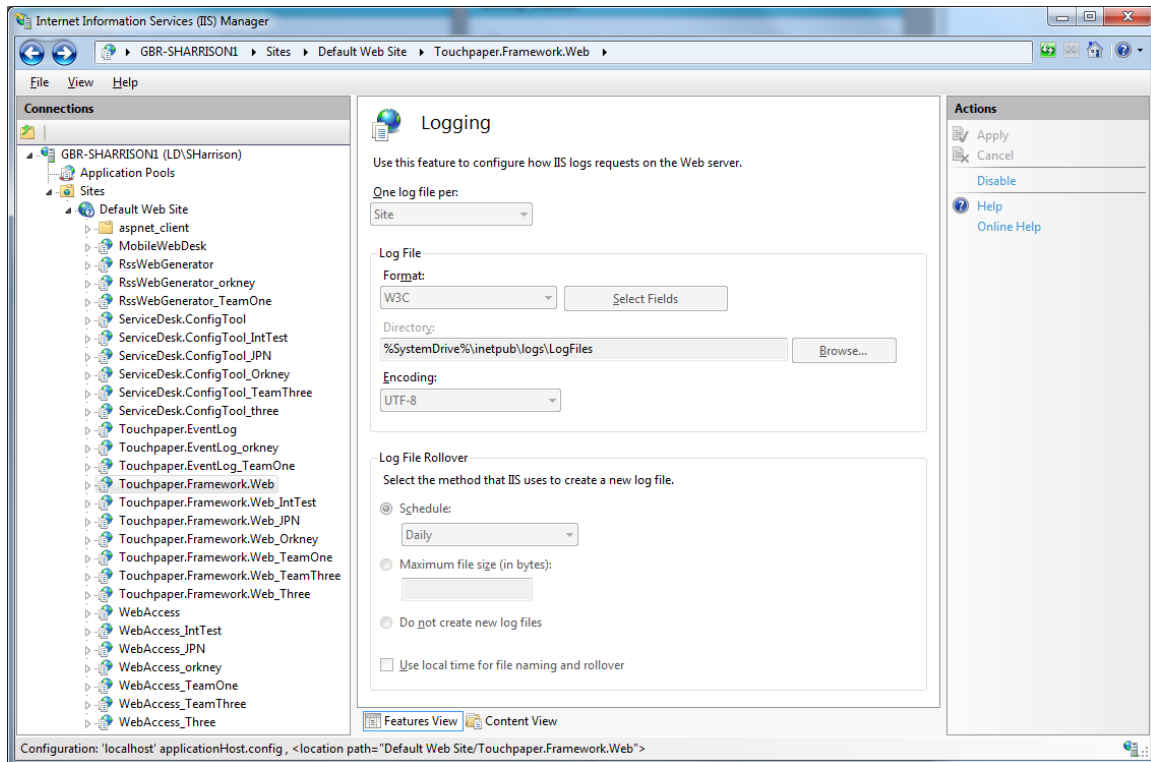
NOTE: For more information about configuring the log severity for the LANDESK Application Server engines, refer to the *LANDESK Service Desk Setup Guide*.

Disable IIS logging

Logging IIS activity is disk intensive. By default, IIS logs all web activity to each website or web application. This is not usually required, so you can consider disabling this within IIS.

To disable logging on a Web site:

1. Start Internet Information Services Manager.
2. Click the required Web site, then double-click **Logging** in the right hand **Features View** pane. The Logging pane appears.



3. In the **Actions** pane, click **Disable**.

Clearing Inbound Mailboxes on the Mail server

Inbound Mail Manager reads through all of the e-mails in the inbound mailboxes each time it polls, so we recommend that you regularly move or archive read e-mails in these mailboxes. Consult with your Mail Administrator on setting this on your Mail server.

Archiving outbound Mail Manager table entries

Outbound Mail Manager creates entries in the tables **tps_user_message** and **tps_user_message_recipient**. These tables can fill quickly and so cause performance problems. A stored procedure is available from LANDESK that archives these entries to a different table. We recommend that if you use Mail Manager regularly, you schedule this stored procedure to run regularly.

Also refer to the following community articles:

MS SQL: <http://community.landesk.com/support/docs/DOC-4949>

Oracle: <http://community.landesk.com/support/docs/DOC-5332>

Service Desk Framework error logging

Make sure that Service Desk Framework message logging is *not* set to all messages (for example information level) and left on, as this can affect performance. Error logging level can be left on permanently as this logs errors only when there are errors and so should not affect performance.

You set the TraceLevel for Service Desk Framework message logging on the Service Desk Web server in the file **C:\ProgramData\LANDESK\ServiceDesk\servicedesk.Framework\tps.config**.

Database maintenance planning

We recommend that a regular backup and maintenance plan is in operation against the Service Desk database.

The implementation of such plans is the responsibility of the customer's Database Administrator.

SQL Server and Oracle: Rebuild Indexes

Regular re-indexing of tables or defragmenting of indexes helps to optimise database performance. Also, monitor database statistics (especially after large data imports). We recommend that DBAs review index fragmentation in the database and if necessary rebuild.

Microsoft SQL Server provides a standard maintenance plan to rebuild indexes on a database. We recommend that you schedule this to run on a weekly basis.

SQL Server and Oracle: Update statistics

Microsoft SQL Server provides a stored procedure called **sp_updatestatistics** that updates information about the distribution of key values in the database. We recommend that you run this stored procedure after importing data. Consider scheduling this stored procedure to run every overnight.

For Oracle you can configure an automated task in Oracle Database Control to gather optimizer statistics during a defined maintenance window.

SQL Server and Oracle: Indexing the database

Indexing a database can significantly improve performance by providing faster searching. However, take care with indexing, because it can affect data writing and deleting because the indexes also need to be updated. Indexing also makes the database larger.

For Microsoft SQL Server implementations, use this only after all other performance options have been explored. For Oracle implementations this is recommended initially and if many design changes are made. Object and Query Designer changes affect the indexes.

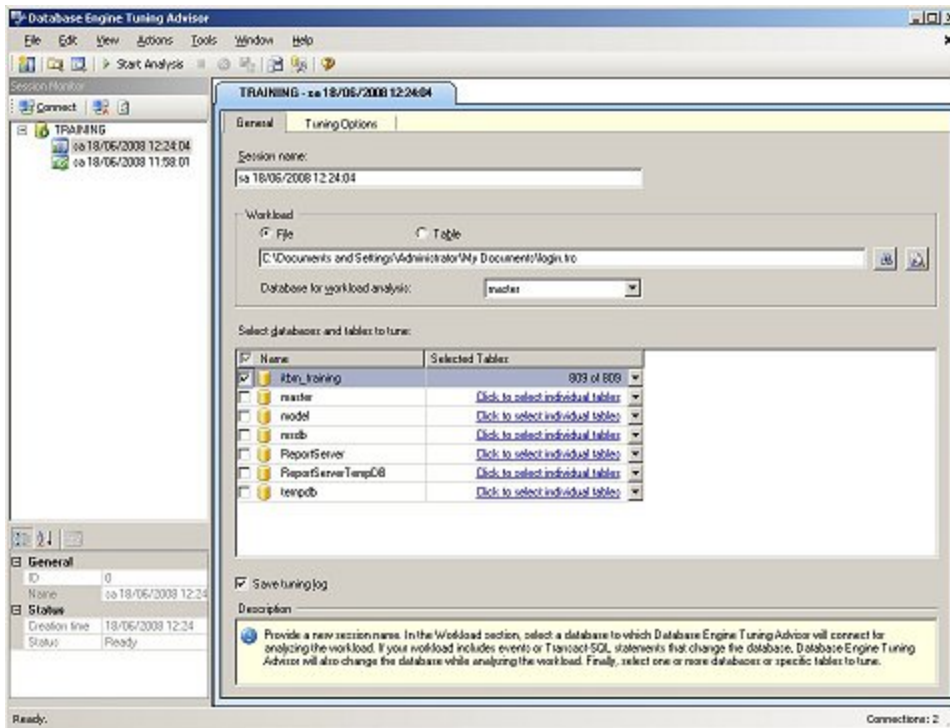
Microsoft SQL Server provides tools that enable you to identify and implement database indexes that will optimise the performance of procedures that are perceived to be slow.

On a test copy of the system, time the procedures, add the indexes using the procedure below, and then time the same procedures again.

To create new indexes in SQL Server:

1. Start **SQL Profiler** and create a new trace.
2. Select the **Tuning** template.
3. Start the trace and perform the non-performing steps that you previously decided on (for example, logging and assigning an incident within Console), timing each stage.
4. When you have finished the predefined steps stop the trace, and save the trace as a trc file.
5. Start the Database Engine Tuning Advisor.
6. In the **Workload** group, select the **File** option button, then browse to your trace file.

7. Select the check box for the database you want to use for the workload analysis – this is just where the tuning advisor does its work; it's best to stick with the same database throughout.



8. Click **Start Analysis**.
The Database Engine Tuning Advisor analyses the database using the trace.
9. When the advisor returns its results, on the **Action** menu, click **Save Recommendations**.

NOTE: If there are no recommendations, then the performance issue isn't likely to do with the database itself.

This script can be run on a copy of your database and will create the necessary indexes.

NOTE: The script may need to be edited. At the top of the script will be: **USE <dbname> GO <dbname>**. This either needs to be corrected to show your database name as it is on site, or the whole "Use, Go" statement can be deleted if you make sure you are running the script on the correct database.

10. After indexing the test database, run through the steps you decided on and take new timings to compare with the original timings. If there is any significant improvement it is probably worth running the script on the live system. If not, it's best NOT to create indexes for the sake of it because of the aforementioned impact on writing data.

Similar tools are available for Oracle such as the SQL Tuning Advisor which can be used to recommend and implement indexing changes.

WARNING: These tools can sometimes advise to remove indexes. Do not remove any existing indexes from the database.

Overcoming issues caused by applying statistics to your database

If you have applied statistics to your Service Desk database (possibly recommended by the Database Engine Tuning Advisor), you may encounter problems when modifying attributes using Object Designer. Some changes to an attribute require the database to drop and recreate a database column, which is prevented when statistics are present.

You can overcome this issue by dropping the statistics from the database, making the change in Object Designer, then reapplying the statistics.

VMWare (ESX)

A number of customers have started using ESX server, which can improve performance because:

- It removes any network traffic if the database and application servers are run on the same ESX Server
- Additional servers can be added to scale out the solution relatively cheaply, quickly and easily

Areas that have been seen to affect performance on an ESX Server environment usually come down to how the server or guest operating system has been configured, or the ESX server id being under-specified to run the number of guest operating systems. For example:

- Configuring a guest operating system for 4 CPUs, but enabling 1 only CPU – effectively making it a single CPU system)
- Under-specifying the network cards (NICs) in an ESX Server. More virtual servers potentially need more network bandwidth and so more network cards.

NOTE: Check with VMWare for their recommendations on configuring virtual machines for best performance.

Reports

This section is equally valid if you run reports in the Service Desk Crystal Reports component or directly against your database.

Running reports that return a large number of rows from the larger tables within your database can temporarily lower the performance of Service Desk for all users, especially if the reports include large text fields.

Ideas to reduce these effects:

- create a database view that returns only the required data for the report and base the report on this. This will extract less data from the database when running the report.
- where possible, schedule the report so that it runs outside of busy periods. For example, Service Desk is supplied with a Crystal Enterprise license that enables you to schedule reports to run outside business hours.
- ensure you follow your report software's recommendations on how to improve performance when designing reports. For example with Crystal Reports, including large text fields within a sub-report rather than in the main body of the report improves the performance of the report and the impact on other users.
- if your reports do not need to be "real time", run the report against a copy of your live database rather than against the database that is currently in use by Service Desk users.

E-mailing query results and reports

Service Desk enables you to e-mail the results of queries to one or more people. For queries containing no prompts this works fine, but if you e-mail the results of a prompted query, the query is executed and ALL data is returned ignoring the prompted criteria. We recommend that you do not e-mail prompted queries as this can cause very noticeable drops in performance on a live system.

Service Desk Console configuration advice

This section contains information about configuration settings that can improve the performance of the Service Desk Console. The most common issues arising when using Service Desk are based around network access between where the console is running and the central site.

Enabling client-side caching

Enabling client-side caching stores local copies of information that is usually retrieved from the server and so can improve performance.

In the **console.exe.config** on the local machine where console is installed there is the following key:

```
<add key="ClientMetadataCaching" value="false" />
```

Change this to:

```
<add key="ClientMetadataCaching" value="true" />
```

The default location for console.exe.config is **C:\Program Files\LANDESK\Service Desk\Console**.

NOTE: See also the following community article: <http://community.landesk.com/support/docs/DOC-4558>

Enabling compression

You can improve performance over a slow network or WAN by enabling the compression option in the local **console.exe.config** file on each computer that is running the Service Desk console.

This reduces the amount of data passed over the network. However, it may have an adverse effect on a LAN owing to the additional processing required at each end to compress/decompress, so use this only on WAN links.

NOTE: Compression can also improve data import and increase the amount of data that can be imported during a single import.

To enable compression:

1. Take a backup copy of **C:\Program Files\LANDESK\Service Desk\Console\console.exe.config**.
2. Add the text in bold below to the file:

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
<configSections>
<section name="microsoft.web.services2"
type="Microsoft.Web.Services2.Configuration.WebServicesConfiguration, Microsoft.Web.Services2,
Version=2.0.0.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35" />
</configSections>
<microsoft.web.services2>
<filters>
<input>
<add type="Touchpaper.Framework.SystemServices.SoapExtensions.CompressionInputFilter,
Touchpaper.Framework.SystemServices.SoapExtensions" />
</input>
<output>
<add type="Touchpaper.Framework.SystemServices.SoapExtensions.CompressionOutputFilter,
Touchpaper.Framework.SystemServices.SoapExtensions" />
</output>
</filters>
<diagnostics>
<trace enabled="false" input="inputTrace.webinfo" output="outputTrace.webinfo" />
</diagnostics>
</microsoft.web.services2>
<appSettings>
<add key="Culture" value="en" />
<add key="TPS Host" value="http://localhost/TPS.411 " />
<add key="Url" value="http://localhost/TPS.411 " />
<add key="IsCompressionEnabled" value="1" />
<add key="ChangeManagement.Change.AddConfigurationItem"
value="Touchpaper.Console.SnapIn.Analyst.AddConfigurationItemHandler" />
```

```

<add key="ChangeManagement.Change.DetachConfigurationItem"
value="Touchpaper.Console.SnapIn.Analyst.DetachConfigurationItemHandler" />
<add key="ChangeManagement.Task.AddConfigurationItem"
value="Touchpaper.Console.SnapIn.Analyst.AddConfigurationItemHandler" />
<add key="ChangeManagement.Task.DetachConfigurationItem"
value="Touchpaper.Console.SnapIn.Analyst.DetachConfigurationItemHandler" />
</appSettings>
<system.web>
<webServices>
<soapExtensionTypes>
<add type="Touchpaper.Framework.SystemServices.SoopExtensions.ExceptionHandlerExtension,
Touchpaper.Framework.SystemServices.SoopExtensions" priority="1" group="0" />
</soapExtensionTypes>
</webServices>
</system.web>
</configuration>

```

NOTE: Note that in addition to the `<configSections>` and `<microsoft.web.services2>` sections at the top of the example, that `<add key="IsCompressionEnabled" value="1" />` has also been added to the `<appSettings>` section.

3. Make sure that Microsoft Web Services Enhancements (WSE) v2.SP3 is installed on each client computer where compression is to be enabled.

NOTE: This can be downloaded from <http://www.microsoft.com/downloads/details.aspx?familyid=8070E1DE-22E1-4C78-AB9F-07A7FCF1B6AA&displaylang=en>

Local Service Desk Framework

In some cases, running an instance of the Service Desk Framework locally to the remote Service Desk can improve performance. Less data may be transferred at the database level than at the Framework level.

Loading Query definitions from cache

You can cache query definitions locally, which can speed the display of results over a slow network. You need the appropriate privileges to configure this.

To enable query definition caching:

1. Within the Service Desk console, on the **Settings** menu, click **System**. The System Settings property grid appears.
2. In the **Query Settings** group, set the **Load Queries From Cache** property to **True**.

Query Settings	
Load queries from cache	True
Default Column size type	Designer Specified
Default Query page size	15
Query Results ViewFilter-Panel Pinned	False
Group Header Appearance	

NOTE: With query caching enabled, any changes to query designs are seen only after restarting the Service Desk console, because the cache is created at logon. This can increase the time taken to log in to Service Desk.

Automatic knowledge searching

Although you can configure a knowledge search to be automatically invoked as analysts are typing in incident descriptions, you could choose not to configure it in this way in all instances, as it can be fairly resource intensive on the server.

If the searching is required, consider how it is invoked. For example, if you have automatic knowledge searching switched on for a large text attribute on your window it would probably be inappropriate to invoke the search on typing of a "punctuation character" as there may be many typed within the text field – each time that a punctuation character is typed a new search of your knowledgebase will be started. In this instance, it might be more appropriate to set the search to start on "lose focus" so that the search starts only once the text has been completed and the user moves to another field. To check the knowledge settings on an attribute, highlight the field in Window Manager and then look at its properties.

In certain circumstances, knowledge searching can be triggered more frequently than has apparently been configured. If the performance of a specific window (typically the Incident window) is still an issue, check that there are no Knowledge Handlers on the window. See *Knowledge searching* on page 26.

Terminal Services

At some point the network link may be too poor to provide good performance of the Service Desk console application. In these circumstances, we recommend terminal services solutions (for example, Citrix or Windows TS) or Web Desk access.


If you use a terminal services solution, we recommend that you enable configurable memory handling for the console client used by terminal services. Add the following lines to the console.exe.config file:

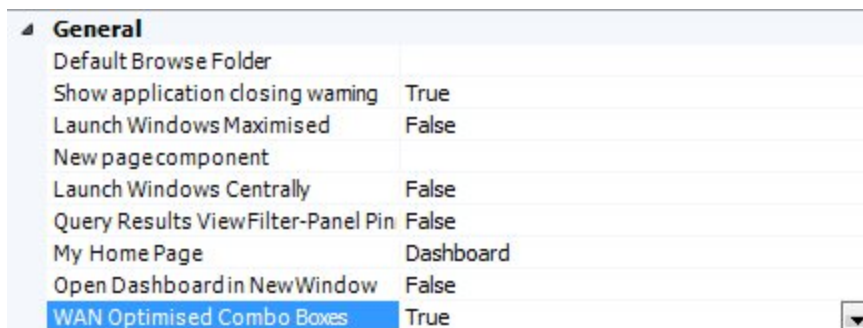
```
<add key="MemoryUsageEnabled" value="true"/>
<add key="MinMemoryUsage" value="2000000"/>
<add key="MaxMemoryUsage" value="75000000"/>
```

WAN Optimised Combo Boxes

When you use a combo box, a request for data is sent to the database each time you type a letter within it. If you use the Service Desk Console over a WAN, this may slow down performance. However, you can configure the client to request data from the database only when you press F4, ALT+down arrow, or ENTER.

To set WAN Optimised Combo Boxes:

1. In the Service Desk console, on the **Settings** menu, click **Personal**.
The Personal Settings property grid appears.
2. Alongside **WAN Optimised Combo Boxes**, select **True**, then click .



Design

In addition to the configuration issues described earlier, there are design choices that you can make that will lower performance. The Service Desk designers provide a wide range of design choices, some of which have little or no effect on the performance of the final system, and some that can have a significant effect on performance. In some situations, it can be appropriate to make design decisions that lower the performance of the system – however, these design choices must be limited in their use and, wherever possible, not used for commonly used areas of the system.

In this section, you can find out about:

- *General observation* on page 18
- *Query configuration* on page 19
- *Query design* on page 18
- *Dashboard design* on page 20
- *Process collections on Group* on page 20
- *Avoiding large collections (unbound collections)* on page 21
- *Configuring queries or filters below the window (tabbed queries)* on page 21
- *After Read Calculations* on page 22

General observation

At every stage of your design consider what you are asking the system to do, and ask yourself if it is likely to reduce the system's performance. If it is likely to affect performance, ask if it is really necessary.

You could create a complicated query that automatically refreshes every five minutes - and this could be a perfectly reasonable design requirement for a particular analyst at a particular time. However, making that query available to all analysts all of the time will impact performance. It is better to design the system so that the query is made available only when it is needed, and only to the analysts who actually need it.

Similarly, if analysts using Web Desk need only a subset of the available fields most of the time, then design the system so that the window displays only the fields they need.

You can improve the performance of a system considerably by considering the performance effects of design decisions at every step.

Query design

Related attributes (for example, User.Title from Incident) added to a query increase the number of database joins that are needed. Adding a large number of these to a query can therefore make the query run slower. Where possible, add related attributes of this type to the query's Preview panel - these are then displayed only when the record is selected in the query results.

Similarly, where possible, add large text fields to the Preview panel where they can be more easily read, rather than to the results list.

Query configuration

When designing queries, set them to launch within their own window, and do not set the query refresh rate too high. If you have a query that is constantly accessing the database because it has been set to refresh frequently (particularly if a lot of analysts are going to be running the query at any one time), this will reduce performance.

General settings specify additional information for creating a query.

General Settings for Query

Title: ...

Description:

Results Detail

Interval in minutes for automatic refresh of results

Include soft deleted records

No. Records Displayed

Return rows per page

Return top rows only

Set this to a high value

Prompt Options

Prompt dialog to use when conditions require user input before producing results.

Prompt Dialog:

Launch Action

You can specify the default action that occurs when an item is selected.

Launch:

Using data from:

Note: You can drag a related Object from the Attributes tree into the text area above.

Launch in a new window when double clicking results

Launch result in workspace if one result

Launch workspace in a new window

Select this checkbox

Audit details

Created By: On:

Updated By: On:

Dashboard design

You can add a large number of queries of any type onto the dashboards: however, we recommend that you do not add more than 2 or 3 queries on a screen. Also note that the more complex a query is, the more resource is required to display the query results. Slow login times can often be attributed to the queries added to a dashboard. Also consider whether the Home page has to be set to the dashboard, or whether it is better to add the dashboard as a component on the shortcut bar, so that it can be accessed whenever required - but not automatically at login.

Only add queries that are actually needed all of the time to a dashboard. Add shortcuts to queries that are needed on a more occasional basis – then they are run only when they are needed. Also, do not set the queries to refresh automatically more often than is necessary.

The dashboard has to run all of its queries each time it is displayed, so include on your dashboard design only queries that are set to launch results in a new window when an entry is double-clicked. In this way the user can return to their dashboard without re-running all of the queries.

Process collections on Group

Collections (one-to-many relationships within the database) can cause performance issues. In particular, there have been instances of systems having Incident, Problem or Change collections on the System\Group object. These collections can have a very bad effect on performance, and we strongly recommend that you remove these collections.

To remove process collections from the System\Group object:

1. Within the Object Designer, open the System branch and display the Group object.
2. If there are any collections present for Incident, Problem or Change delete them.



Avoiding large collections (unbound collections)

The best way to avoid large collections is to make sure that your class model does not include any 'unbounded' collections. An unbounded collection is any collection that is likely to grow as the size of the database grows.

For example, Incident-Notes is a bounded collection because one would normally expect an Incident to be open for a finite period of time and for the number of Notes to be limited by the amount of information attached to that one Collection. On the other hand, a Category-Incidents collection would be unbounded because you would expect the number of objects in the collections to increase as the size of the database, ie the number of Incidents logged for each category grows.

Whenever you create a relationship in Object Designer ask whether it is likely to lead to an unbounded Collection attribute. If so, then make the relationship one-way.

NOTE: A script is available from LANDESK support that checks for unbounded collections.

NOTE: See the following community article for further information:
<http://community.landesk.com/support/docs/DOC-22310>.

Configuring queries or filters below the window (tabbed queries)

You can add and modify queries or filters in a tab below a window in Window Manager. This can provide your analysts with useful information such as what other incidents have been logged by a particular user, or what CIs are associated with the user. However, it is important not to overload the system with too many resource-hungry queries. Every time an analyst updates an incident they will be running the tabbed query or queries associated with the incident window. If the query is complex it may take some time to return data on the window.

If you want to have these queries or filters on tabs below a window, DO NOT add them to the first tab - as this tab is always displayed. The data for subsequent tabs is not retrieved until the tab is clicked (unless the queries are set to auto-refresh).

The data for queries that are set to auto-refresh is also ALWAYS retrieved when the window initially displays, and then is retrieved again after the refresh interval - even if the tab is not clicked. As a result of this, DO NOT add queries or filters that have auto-refresh settings to windows.

General settings specify additional information for creating a query.

General Settings for Query

Title: All Incidents

Description: All Incidents

Results Detail

Interval in minutes for automatic refresh of results: 0

Include soft deleted records

No. Records Displayed

Return 6 rows per page

Return top rows only

Prompt Options

Prompt dialog to use when conditions require user input before producing results.

Prompt Dialog: Simple List

Launch Action

You can specify the default action that occurs when an item is selected.

Launch: Workspace

Using data from: Incident

Note: You can drag a related Object from the Attributes tree into the text area above.

Launch in a new window when double clicking results

Launch result in workspace if one result

Launch workspace in a new window

Audit details

Created By: SA On: 08/07/2008 14:19:17

Updated By: SA On: 08/08/2012 13:50:56

Finish Previous Next

For queries added to windows, set this to 0...

...and this to a low number

Also, limit the number of records that are displayed per page for these queries, as this will improve performance, and as these queries are being displayed below a window there will be less space for them to display a large number of rows anyway.

After Read Calculations

Avoid running large queries that contain After-read calculations, because a calculation is performed on each row that is returned by the query. If there are a lot of calculations to perform this can add an overhead on to the query.

Analysing a poorly performing system

This section provides hints and tips to help you analyse and resolve a poorly performing Service Desk system.

CAUTION: The most important piece of advice is to ensure that the environment where Service Desk is being installed fulfils the requirements of the *LANDESK Service Desk Supported Platforms* document and the *LANDESK Service Desk Technical Specification & Architecture Guidelines*. Both of these documents are available from LANDESK.

General analysis steps

Witness the reported performance issue yourself, don't accept someone else's description: you might see something different.

Identify the architecture and what's running and where it's running: draw a diagram. Ensure that all of the servers have sufficient free disk space.

You need to get an understanding of areas where you believe performance needs to be increased and where it is already working well. For example:

- Is the system slow with a single user or does it just slow down with multi user access? If it's slow with one user then the issue is probably some sort of configuration/environment issue.
- Is the system slow when logging in / loading windows / loading certain windows / running queries / running certain queries?
- Is the system slow when using Console or Web Access or both?
- Does the system run faster when on a client computer or on the server itself?
- Does the system run well for periods in the day or when no active users are in the system but slow periodically or in periods of high usage?

Use performance monitor to monitor the server specifically for the Service Desk Framework (for both Console and Web Access) and for Web Access.

Monitor the W3WP, W3WP#1 ... processes for CPU activity and private bytes. Is the average CPU of the processes high (70%)? Are private bytes increasing without coming back down (no saw tooth graph?). If the Service Desk Framework and Web Desk are running in different application pools, there will be a different W3WP process for each. The name of each W3WP process will depend upon which started up first. For example, if the first connection was to Web Desk then W3WP will be Web Desk and W3WP#1 will be the Service Desk Framework. If the first connection is Service Desk then W3WP will be the Service Desk Framework. This will get more confusing if other applications (and application pools) are running.

Use Windows Task Manager to identify if other applications are running that are taking up resources (CPU or memory for example).

Error logging – are there errors in the event log? Turn on Service Desk Framework logging to record errors. Not only will this give more detail but it will also indicate which users are generating errors, which may be useful in tracking down the problem. Sometimes a slow performing system comes down to errors.

Check the timings of specific operations (for example, displaying windows, running a query, saving an incident) and compare these results with the expected results.

Is performance a general issue or in a specific area?

Set up a test system.

Run Console and Web Desk on the server that is running the Service Desk Framework and Web Desk – is this different from the end-users' experience? If so, this may mean it's a network or desktop issue.

Service Desk Design Issues

Are there certain specific designs that have been implemented that are causing specific performance issues? For example:

- Queries that produce inefficient or slow SQL – this can be analysed with profiling tools or recording specific queries using the Service Desk Framework logging features and passing these queries through optimisers and profiling tools.

Service Desk bugs

Some performance issues may be down specifically to bugs. This will normally relate to handling large amounts of data inefficiently. For example a bug found (and fixed and patched for 7.2.3) revolved around adding end users to groups and roles in the administration module where there were already a large number of users in that particular group or role. These types of issues are usually reasonably easy to track down as long as a good description / analysis is provided by the user.

Database

The main things to look for are database timeout errors occurring in the Service Desk Framework error log and a large number of deadlock errors as well (deadlock errors will always occur, but may be more frequent in a poorly performing system). If these are happening it usually means that some indexes on some tables have become fragmented. Fragmentation occurs when large numbers of records are added to tables. This can happen gradually over time (for example, over a year as the number of incidents increases) or quickly if a large import is done. The solution is to re-build the indexes on all tables. We recommend that this is done periodically (for example monthly) or after a large import. In SQL Server you can set up a database maintenance plan to perform this regularly. Oracle has its own tools to perform these tasks as well, such as Oracle Scheduler.

There may be specific SQL queries that are taking a long time irrelevant of any fragmentation. You can analyse these as described previously using profiling tools and Service Desk Framework diagnostic logging.

NOTE: See the Appendix in the *LANDESK Service Desk Administrator Guide* (LDSAdministrator.pdf) for details of how to use Service Desk diagnostic logging for SQL tracing.

The database server may be being shared with other databases that may be affecting the performance of Service Desk. This should be identified and if possible the Service Desk database moved to a dedicated server.

Check the database for bespoke work such as triggers and stored procedures.

Service Desk

Ensure that communication is not going via a proxy server (LAN or WAN). You can check this using Internet Explorer's LAN Settings from the Connections tab under Tools\Options. If messages are going via a proxy, then they may be going further distances and through a poorly performing proxy server. This will slow down the communication (high latency) so reducing performance. Exceptions can be set up so that the proxy server is by-passed.

When you connect to a Web server using the Internet Protocol (IP) address or Fully Qualified Domain Name (FQDN) on the local network, the connection is through an assigned proxy server even if the Bypass proxy server for local addresses option is enabled.

However, if you connect to a Web server using the host name (for example, <http://webserver>) instead of the IP address (for example, <http://10.0.0.1>) or FQDN (for example, <http://webserver.domainname.com>), the proxy server is bypassed and Internet Explorer connects directly to the server.

NOTE: For information about bypassing a proxy server, see:
<http://support.microsoft.com/kb/262981/en-us>

NOTE: You can add proxy exclusions directly to console. For more information, see the following community article: <http://community.landesk.com/support/docs/DOC-5161>.



Use local file caching – for LAN or WAN access. This caches meta data locally in files, so once created subsequent start-up of the application and initial actions should be improved (7.2.4 and above only).

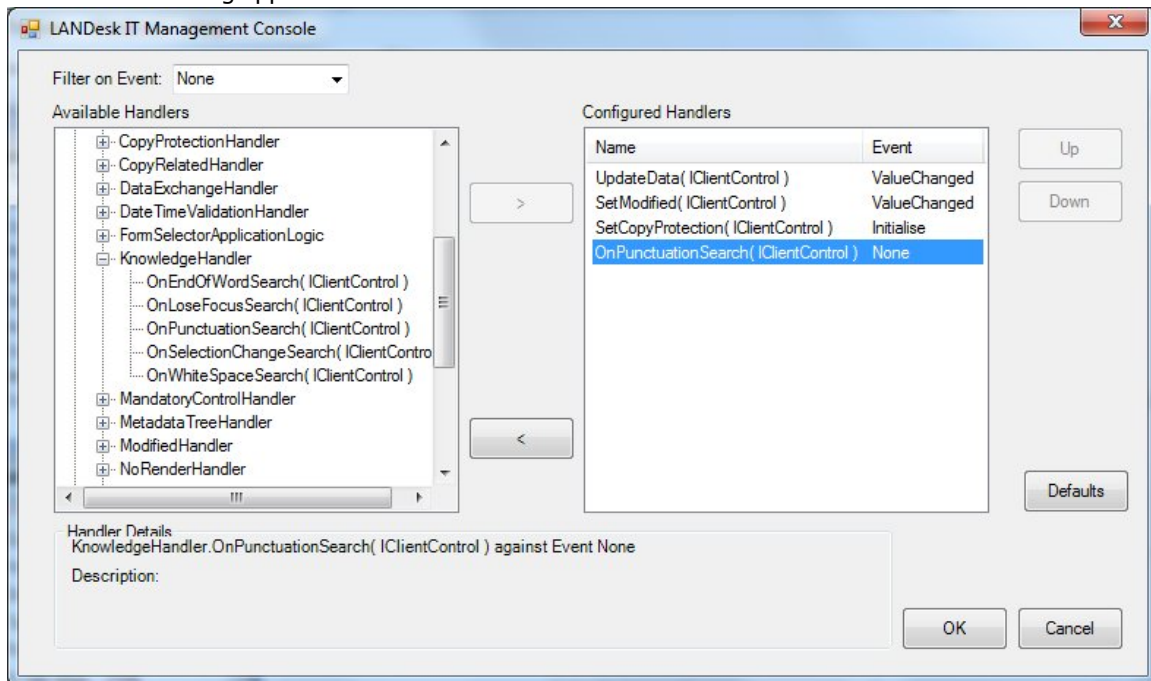
If performance is poor under load, then it is likely to be due to the load being placed on the instance of the Service Desk Framework that all Consoles are connecting to. Check this using performance monitor. If this is the case, then the instance of the Framework will have high CPU utilisation. In this case scale out of the Service Desk Framework Web server is required or add additional CPUs to the server. If scale out is chosen so running multiple servers, load balancing can be implemented over the multiple instances of the Framework or Service Desk applications can be 'hard wired' to specific Framework instances. If the number of concurrent Service Desk users is around 100 for each Framework instance this may likely be the cause of the performance problem.

Knowledge searching

In certain circumstances, knowledge searching can be triggered more frequently than has apparently been configured. If the performance of a specific window (typically the Incident window) is still an issue, check that there are no Knowledge Handlers on the window.

To check for the presence of Knowledge Handlers on a window:

1. Make sure that the keys `<add key="ShowHandlerConfiguration" value="true" />` and `<add key="ConfigureHandlers" value="true" />` are added to the `console.exe.config` for the client that you are using, then start Service Desk.
2. Start the Window Manager component, and open the required window.
3. Click  to display the **Properties** grid, then click a control on the window.
4. In the Properties grid, under **Configurable Properties**, click  alongside **Handlers**. The Handlers dialog appears.



5. Make sure that none of the **KnowledgeHandler** handlers are configured for the control.
6. Repeat this for each control on the window, then save the changes.

Other applications

Other applications running on the same computers may have an effect on the performance of Service Desk applications. For example, Virus Scanning software has been known to cause problems. Virus scanning software is not a problem in general, but issues have been seen where versions of software were incompatible with the hardware they were running on. This had the effect of a very poorly performing web application with a single user. On analysis, the CPU profile in this case was very different from what was expected. Rather than seeing a spike of CPU per user operation, the CPU usage was more of a 'tower' shape, ie more CPU processing was being used to perform the same operation. In this case this was shown up in the processing of the web application itself not the virus scanning software presumable due to the nature in which the virus scanning software 'attaches' itself to applications.

In most cases virus scanning software is compatible with Service Desk applications, this was a special case. However, due to the nature of virus scanning, performance of the application will degrade (hopefully only slightly) with virus scanning in use. One way to eliminate this from the problem is to turn virus scanning off temporarily to see if it has any significant effect on the overall performance. If so, then there is a problem.

Interaction with other applications can be the hardest issues to analyse.

Hardware issues

Just because the system is not on the minimum specification recommended by LANDESK does not mean this is the cause of the performance problem. Normal analysis such as running performance tools to monitor CPU and memory will indicate if this is actually the problem.

Specifically check for:

- Memory utilisation
- CPU utilisation – Dual or quad will give better performance under load
- Network Card – Has it been configured to 10Mbits/s rather than 100Mbits/s
- Network Routers – Are they configured to prioritise between different types of traffic i.e. http
- Network speed – connection from client to server, network hops, network traffic

Checklist

The following is a quick checklist to run through if you are finding performance issues:

Database

- Clear down the tps_user_message table regularly.
- Archive or clear down outbound Mail Manager entries.
- Check for unbound collections: <http://community.landesk.com/support/docs/DOC-4955>.
- Check whether Snapshot Isolation is enabled on SQL Server.
- Update statistics.
- Rebuild indexes on all tables regularly.

IIS

- Turn off Error logging on web sites.
- Turn off Log Site Visits on web sites.
- Consider splitting Service Desk Framework web sites for multiple users.
- Consider separate Service Desk Framework for Services on Application server.
- Ensure separate Application pools for Service Desk Framework and Web Access web sites.
- Set Application pools to recycle each night (or period of low usage) or not at all.

Servers

- Check specification against recommended technical specification guidelines.
- Check disk space (at least 10% free space on OS drives).
- Check memory and CPU usage during poor performance.
- Check Page file size (virtual memory) is set to at least the recommended size in Performance Options.
- Check the event logs for errors.
- Check for other applications that are running (virus scanners, etc). Consider testing with these switched off.

Network

- Check network traffic.
- Check Network Card (speed, usage etc).
- Check routers.
- Make sure a proxy server is not being used.
- Check ping timings from client to web server and web server to database server.
- Check the number of hops as above using tracert to make sure access is short and direct.

Service Desk Framework Configuration:

- Ensure Diagnostic logging is switched off in all tps.config files.
- Consider disabling Process Workload business object.
- Increase timeout (see the Appendix in the *LANDESK Service Desk Setup Guide*, LDSSetup.pdf).
- Try using IP address in connect string rather than name.

Service Desk console

- Consider using client side caching for console.
- Consider enabling compression for console if running over WAN.
- Consider loading query definitions from cache.
- Check if attributes on Incident windows set to 'Invoke Search' and consider changing setting or switching off.
- Check that tabs on Incident windows are not running large queries.
- Check dashboards do not contain lots of queries and that queries do not refresh.
- Check if large reports or queries are being run during periods of slow performance.

VMWare

- If you are running servers on VMWare, refer to the performance guides for VMWare Server.
For example:
http://www.vmware.com/pdf/vi_performance_tuning.pdf
http://www.petri.co.il/virtual_increase_vmware_performance.htm
- Poor memory configuration in VMWare is a common cause of poorly performing applications.

