NotesTracker

for IBM® Lotus Notes® / Domino®

Version 6.05

Administrator Guide

https://www.agecom.com.au
NotesTracker Support: notestracker@agecom.com.au

CONTENTS

ABOUT THIS GUIDE

This guide is for Lotus Notes / Domino Administrators who are involved with the deployment of NotesTracker and the management of applications in which NotesTracker has been implemented. It provides detailed information on deploying NotesTracker, setting up & configuring NotesTracker settings, and setting up alerting.

This guide is also recommended for Notes developers to enable them to become familiar with NotesTracker administration options.

The first step to implementing NotesTracker is for your Lotus Notes / Domino developers to add the necessary NotesTracker design elements to your applications. The steps to perform this are covered in the NotesTracker Developer Guide.

After NotesTracker has been installed into your applications and setup / configured the NotesTracker User Guide provides information on viewing NotesTracker logs, analyzing data, and creating reports.

DISCLAIMER

This guide could include technical inaccuracies or typographical errors. Please advise us if you see the need for any corrections. We welcome your feedback about any technical inaccuracies plus your comments about this guide’s clarity, usability and missing or incomplete information.

Changes are periodically made to the information herein. These changes will be incorporated in new editions of the publication. For all but trivial changes (such as minor errors in spelling, punctuation and layout) the modified publishing date will be displayed on the front cover.

AGE Computer Consultancy (herein referred to as *AGECOM*) may make improvements and/or changes in the software and its features described in this guide at any time without notice. You should periodically check our website for notification of such changes.

WHAT IS NOTESTRACKER?

NotesTracker is a comprehensive ***application enabling*** developer toolkit. The application makes it easy and simple for you to extend your IBM Lotus Notes and Domino database applications so that you can rapidly detect and report on a wide range of actions being carried out on the content of the databases.

NotesTracker was originally developed by Asia/Pacific Computer Services and is now proudly provided and supported by AGECOM (https://www.agecom.com.au).

NotesTracker monitors the creation, reading, update, and deletion of documents in Lotus Notes databases.

Furthermore you can track not only that a particular Notes document in its entirety was updated (when, where, and by whom) but what parts of a document's content was altered. This is done by your selecting the logging of the “before” and “after” values stored in all fields or just user-nominated fields in the document.

You can also track changes to *attachments* stored in documents, by logging the “before” and “after” names and sizes of attachments. The attachments themselves are not logged, since this could enormously increase the Repository database's disk space requirements, with attachments typically being megabytes in size. NotesTracker has been designed and coded carefully to run as efficiently as possible in order not to add a significant performance burden to your Notes/Domino installation.

NotesTracker can log other content manipulation actions too, such as the *pasting* of documents into a database or the *mailing-in* of messages arriving from the Internet.

NotesTracker goes beyond just the *passive* logging of actions occurring against database contents. In terms of being *proactive*, NotesTracker empowers your developers to extend your applications by tracking additional types of actions (such as button presses), to generate mail alerts when document content is acted upon in ways that are important to you, and a way for you to build dynamic Notes views for continuous monitoring of content- related events in the RSS fashion that is so popular on the Internet.

For information on implementing NotesTracker into your applications please refer to the NotesTracker Developer Guide.

EXAMPLE USES OF NOTESTRACKER

Here's a small selection of the ways that you might benefit from the use of NotesTracker:

- ✓ **Generate comprehensive document history trails** (sometimes referred to as "breadcrumbs") – See who Creates, Reads, Updates, and Deletes individual Notes documents, as well as document pastes and mail-ins. This is great for troubleshooting, and for finding “the guilty ones” who entered bad data or deleted key documents!
- ✓ **Analyze all the active users** of your Notes databases, arranged to show the **significant users (that is,** "the makers" -- those who contribute most to your applications by adding/updating content – versus "the takers" who consume content but don't develop and maintain it. This can be used to reward your key knowledge contributors, or to nudge less active users.
- ✓ **Field audit trails** – Determine who made what **changes to individual field values** in each document. This could generate a payback by enabling you to gain understanding and content control for as little as a single critical database of yours. (Indeed, one user justified purchase of a NotesTracker license by gaining control over just their Domino Directory database.)
- ✓ Discover what sort of **document content** is most **popular,** or **"hottest"** (you will see the percentage usage of each document in a database). Use this to determine if an application is being used as you intended, whether it needs design improvement or enhancement, and so on.
- ✓ Determine which documents have **low usage rates**, and thus are good candidates to **be archived** or **purged**.
- ✓ **Mail Tracking** – NotesTracker can be applied to any sort of database, but one popular application is the tracking of activities occurring in users' Notes Mail databases – typically for compliance, data secrecy or privacy reasons (e.g. when a user's mail is delegated to any other person).
- ✓ **When and where did it happen?** Examine the date, time and server for each "hit" if needed, once a database's design has been enhanced by adding NotesTracker code. Identify **usage patterns and trends** – over a period of hours, days, weeks, months or years, and in easily-understood categorized Notes views – equally valid for Web- based Domino applications or Notes Client accesses.
- ✓ **Determine detailed user navigation patterns –** How and when each user shifts from document to document, and database to database. Are they using your Notes applications as designed or expected? Does documentation need to be improved or additional application training need to be carried out?
- ✓ Application performance effects? Are specific document actions (such as updates) contributing most to **Domino server loading?** This could give you pointers, otherwise unobtainable, about how you might redesign or reconfigure your Notes applications to lessen server load and improve responsiveness.
- ✓ **Breaking News** – design a Notes view filtered view to display the very newest documents in the order they are created or updated. This is just like generating your very own **RSS-style newsfeeds.**
- ✓ Track **"special" documents** – certain documents in a database that you designate as being particularly interesting and/or important for some reason. These could be any documents. For example, in a Domino Directory database there will probably be Server Documents for a select few of your servers, such as mail hubs, that you want to focus on and be automatically alerted if key fields relating to network performance or security are altered.

- ✓ **Activity Alerts** – get notified via e-mail as soon as documents are updated, deleted, and so on (even when certain fields that you select are updated), or as soon as a document that you nominate as "special" is used. It could be just about anything, such as a sale being won or lost, a credit limit being exceeded, a document is deleted, a special menu or seating request in a travel request application.
- ✓ Track **"generic" events** – this is a NotesTracker term referring to events like button clicks and otherwise untrackable application actions that are of interest. With minimal database design change, you can track a range of application-related actions that might otherwise require lots of coding and testing.
- ✓ Why track everybody? If you so need or desire, you can **track only specific users** of a particular database (or ignore them and track everybody else). For example, you may wish to focus your tracking on the actions of certain key users of a database, or perhaps even just a single user suspected of defalcation or data theft or privacy violations, or only those people encountering problems in using an application, or whatever the case might be.

THE PACKAGING OF NOTESTRACKER

NotesTracker comes in the form of an ‘Installer’ application which provides the following features:

- Propagation of NotesTracker design elements into your applications.
- Upgrade of existing NotesTracker design elements in your applications to the latest version (when updates to NotesTracker are released).
- Creation of a standalone **Repository Database** – referred to as the “NotesTracker Repository” or the “NotesTracker log”.

The NotesTracker Repository database delivers several capabilities:

- It can be used to hold an **audit trail, or log, of the application activity** for your Notes databases. A collection of *usage log entries* builds up over time as the databases' documents are being created, accessed and deleted and when other trackable events occur (such as database or view opens, document pastes and mail-ins, plus others that you can define).
- The NotesTracker Repository database provides, via Notes views, a continuously-updated **analysis of document and view usage patterns** in your own application databases.

  You may have one or more NotesTracker repositories located throughout your Notes/Domino network, and they will behave and replicate just like any other ordinary Notes database. This is important, because it means that no special repository or reporting infrastructure needs to be learned about, installed and maintained – that is, with NotesTracker it’s a simple case of “business as usual.”

  This guide explains how you would decide about and set up a single repository or multiple NotesTracker repositories, each repository used for logging the activities occurring in just a single database (if it’s an important one) or in groups of related databases.

  The NotesTracker Developer Guide discusses the advantages (and disadvantages) of a further option: modifying the design of a given database so that NotesTracker’s usage logging entries are stored within the database itself, rather than getting sent to a central NotesTracker repository.

- The distributed NotesTracker repository database also doubles up as the *code container* for the NotesTracker Software Development Kit (SDK), that is, as the **NotesTracker design repository.**

For more information on using the NotesTracker Installer please refer to the '*Implementing NotesTracker in Your Environment*' topic further in this guide.

EVALUATION AND LICENSED VERSIONS

As a **licensed user** (purchaser) of NotesTracker you will get a version of the NotesTracker Installer containing all NotesTracker design elements (forms, views, agents, script libraries, etc) in which the source code (LotusScript and Formula Language) is available. The design of the NotesTracker Installer is always hidden however you will be able to view the source code of the NotesTracker design elements that have been propagated to your applications or in the standalone NotesTracker Repository database created from the Installer.

Your Notes developers and administrators will be able to easily adapt your Notes databases for comprehensive usage tracking, by building the NotesTracker code into the database designs.

If you are using the free **evaluation** version, all of the functions work without limitations and can be fully exercised, but you do not get access to the underlying NotesTracker source code and design elements.

Also available with NotesTracker (as a separate download from the AGECOM website) are several representative Lotus Notes databases to give you as a potential user some representative NotesTracker-ready applications for familiarization and experimentation with NotesTracker configuration, usage and reporting:

NotesTracker Discussion on Local
NotesTracker Domino Directory on Local
NotesTracker Phone Book on Local
NotesTracker Customer DB on Local

The example databases in the NotesTracker evaluation version are fully functional (but their designs are hidden). Our expectation is that after a little experience using them you will quickly come to see the potential for NotesTracker in your organization and be able to make informed purchase decisions.

The example databases in the NotesTracker purchased (licensed) version do not have their designs hidden. As a licensed user, your Notes/Domino developers should examine the various NotesTracker techniques in these databases as a kick-off point to quickly discover how NotesTracker can be put to use.

This guide together with tips on the NotesTracker News Blog will quickly show your Notes developers and Administrators how to adapt your own Lotus Notes applications and take full advantage of NotesTracker's capabilities.

ABOUT NOTESTRACKER

DERIVATION OF THE NAME 'NOTESTRACKER'

Items such as documents and views in Lotus Notes databases are sometimes referred to as “design notes” or just “notes”, hence the name “NotesTracker” because it is the user actions being performed against these items that are being tracked. NotesTracker’s emphasis is on *application content and activity against it* (rather than on what some other tools track, such as Domino server performance and disk usage, or networking traffic loads).

NOTESTRACKER TRADEMARK, AND FAIR USE

The term “NotesTracker” is a trademark of AGE Computer Consultancy.

NotesTracker is a proprietary product of AGE Computer Consultancy. The information in this guide is **Copyright by AGE Computer Consultancy** but the information is not hidden. The guide resides in the public domain (as a Web download) so “fair use” of the information is allowed.

NOTESTRACKER USAGE TERMS AND CONDITIONS

Our aim: to provide you with a reliable, resilient, capable and low-risk solution backed by professional support, while retaining our precious intellectual property rights invested in NotesTracker and receiving an adequate return in order to be here to support you in the long run!

As a convenience, on the following pages is a snapshot of the NotesTracker terms and conditions of use that applied at the time that this guide was last updated.

Please note that the terms are varied from time to time and you should always obtain latest version of the terms and conditions from the AGECOM website (since it is these which apply to your use of NotesTracker):
https://www.agecom.com.au/NotesTracker

SOFTWARE USAGE TERMS AND CONDITIONS

AGECOM reserves the right to make modifications to these terms and conditions from time to time. The latest terms and conditions are available on the AGECOM web site for viewing prior to the downloading and installation of the Software and will take precedence over any other form of the terms and conditions. Such modifications shall be effective immediately upon posting of the modified agreement at this Web site. You agree to review the agreement periodically to be aware of such modifications and your continued use and/or downloading of software available to you hereunder shall indicate your acceptance of the modified agreement.

COPYRIGHT AND LICENSE

The "NotesTracker" Software (also referred to as "AsiaPac Document Usage Tracker" and / or “AGECOM Document Usage Tracker” Software) is owned by AGECOM and is copyrighted and licensed not sold. The term "Software" means the original program and all whole or partial copies of it. A Program consists of machine- readable instructions, its components, data, audio-visual content (such as images, text, recordings, or pictures), and related licensed materials.

The entity that purchases the license ("you") and all its officers, employees, contractors and other associated persons will be subject to this license agreement.

The **evaluation form** of the Software is provided with hidden design and you shall not modify, translate, disassemble, decompile, or reverse engineer the Software. The **licensed form** of the Software is provided with unhidden design so that its design elements may be incorporated into the designs of your Notes databases either without modification or modified and adapted by you to meet your specific usage metrics requirements. For both forms you may not remove, alter or deface any proprietary notices on the Software.

You acknowledge that AGECOM owns all right, title and interest in and to the Software or portions thereof, including without limitation all Intellectual Property Rights. "Intellectual Property Rights" means any and all rights existing from time to time under patent law, copyright law, trade secret law, trademark law, unfair competition law, and any and all other proprietary rights, and any and all applications, renewals, extensions and restorations thereof, now or hereafter in force and effect worldwide.

The Software and documentation are proprietary to AGECOM. By installing and using the Software, you agree to comply with these terms and acknowledge that the Software and documentation contain valuable trade secrets and other proprietary information belonging to AGECOM that is confidential between you and AGECOM. You also agree to not remove, obscure, or alter AGECOM's copyright notice, trademarks, or other proprietary rights notices affixed to or contained within the Software.

You may copy the Software only for purposes of backup including multiple archive backup copies or incorporation into the design of your Lotus Notes databases provided that each and every copy must contain all of the original Software's proprietary notices. You may not distribute (for free or for sale) or sublicense the Software. You agree to hold the Software in confidence and undertake not to pass copies of it whole or in part to another party outside your organization. If you distribute, rent or sell Lotus Notes applications to other organizations you will not include the Software or any derivative or adaptation of it in these Notes databases until such time that the receiving organizations have purchased the appropriate NotesTracker license from AGECOM.

You will ensure that anyone who uses your copy of the Software does so only for your authorized use and in compliance with the terms of the license. Unlicensed evaluation software may not be used for production purposes.

You are responsible for communicating the terms of this agreement to your employees and contractors and for ensuring their compliance with the terms of this agreement and any company policies and procedures you might have surrounding use of the Software. You must report to us, as soon as possible after you notice it, any suspected misuse arising from your possession of the Software including counterfeiting, piracy, disclosure to non-licensees or other copyright infringement in the Software.

AGECOM grants you a non-exclusive license to use the Software when you lawfully acquire it. When you purchase the Software a License Document will be sent to you by AGECOM as a Proof of Entitlement and this will contain a License Number applicable to the Version of the Software as of the date of purchase, being the date that payment funds are received by us. You may use the Software for evaluation purposes but not in production until the license has been issued. The license will apply to the latest Version of the Software at the time of purchase and to any sub-releases of that Version but not to any subsequent Version if an Upgrade Price applies to the later Version. Your license is void without registration of required information or registration with incorrect information.

AGECOM cannot and does not give credits or refunds for charges already due or paid. Prior to purchasing the Software you are expected to make every effort to establish that the Software meets your requirements. AGECOM will provide you with information and guidance to aid your efforts but the final decision is yours.

The Software is licensed as stated above. The license does not constitute ownership of the Software, only the right to use it. Licenses are not transferable.

If you do not agree to or cannot comply with any of the terms and conditions, do not attempt to access or use the Software. By installing the Software you agree to these terms and conditions. If you decide not to purchase the Software or on expiry of a lease to the Software you agree to destroy all copies including backup copies in electronic or any other form. AGECOM may terminate your license if you fail to comply with the licensing terms and conditions. You agree that you will not continue to use and will delete the design and code of the Software or any derivative or adaptation of it in any of your Lotus Notes databases or elsewhere as soon as practicable after the license expires.

LIMITED WARRANTY

This license entitles the user to hold the source code of the Software in escrow and to run the Software, but not to disclose or sell it to other parties either in its entirety or any individual components of its design and code including internal documentation.

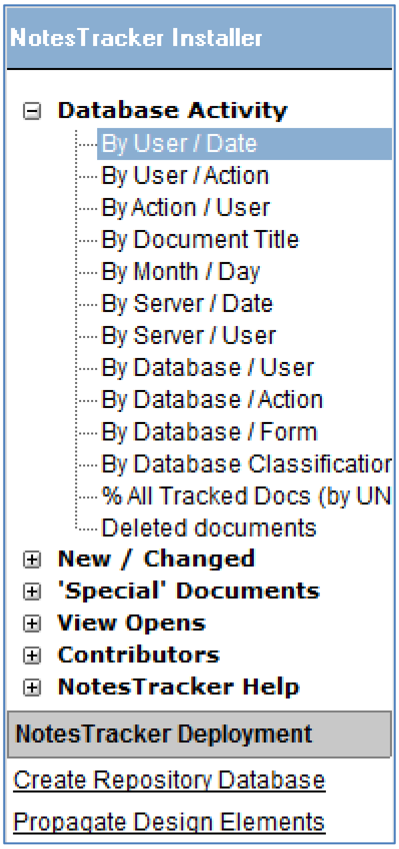
The cost or guarantee of support is not included in the license except that AGECOM will make reasonable effort to fix any problem in the original Software and there will be no charge for fixing any reported problems. However AGECOM reserves the right to charge for enhancements or new versions and for all repair or maintenance of the Software arising from your modification of the original design or code of the Software. The Software is designed to be applied without change to the design of your organization's Lotus Notes databases but you are free to modify the Software and adapt it as you see fit for incorporation into your databases however any such modifications or adaptations and the effects pursuant to them are not part of free support by AGECOM under the warranty.

Subject to any statutory warranties which cannot be excluded, AGECOM shall not be liable for material, equipment, data or time loss, caused directly or indirectly by proper or improper use of the Software or for the effects of any modifications that you make to the Software. In cases of loss, destruction, or corruption of data, AGECOM shall not be liable. AGECOM does not take any other responsibility. AGECOM makes no other warranty.

IMPLEMENTING NOTESTRACKER IN YOUR ENVIRONMENT

The first step to implementing NotesTracker is to open the NotesTracker Installer application. If you haven't already done so it can be downloaded from the AGECOM website at: https://www.agecom.com.au/notestracker.

When the NotesTracker Installer is opened the following options are shown on the navigation menu:



The list of options on the navigation menu provide a typical example of the various NotesTracker views available in the NotesTracker Repository database or which you may implement in your own applications.

The important options to note are those under the '*NotesTracker Deployment*' heading which are used for implementing NotesTracker in your environment. Please note these options are only available in the licensed version of NotesTracker. These are covered in detail below however here's a brief explanation of what each of the options are used for:

- **Create Repository Database**
  This option is for creating a standalone NotesTracker Repository database where NotesTracker logging entries will be written to.

- **Propagate Design Elements**
  This option is for adding NotesTracker design elements to your applications as well upgrading previously propagated design elements to the latest version when an update of NotesTracker is released.

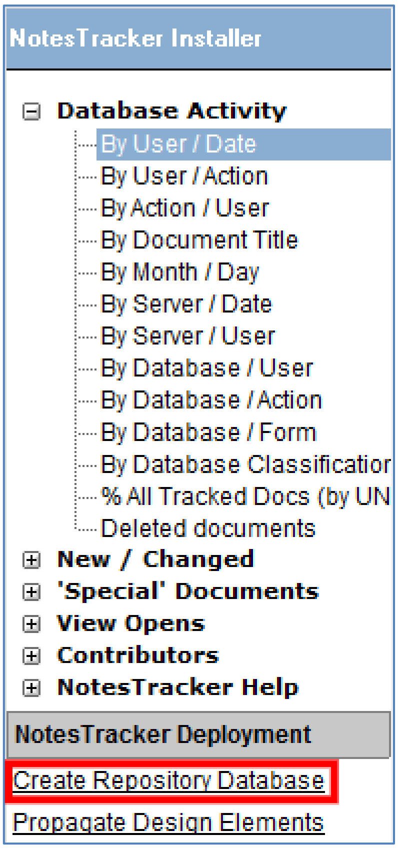
The source code for the various NotesTracker design elements is provided for use by your Lotus Notes developers as the foundation for extending the design of those of your Notes database applications for which you want usage tracking to be performed.

How quickly and easily this can be done will vary from database to database, depending on the database's design complexity (and how familiar your developers are with its design). Many will be adapted in just minutes, or tens of minutes, but some complicated applications might present more of a challenge.

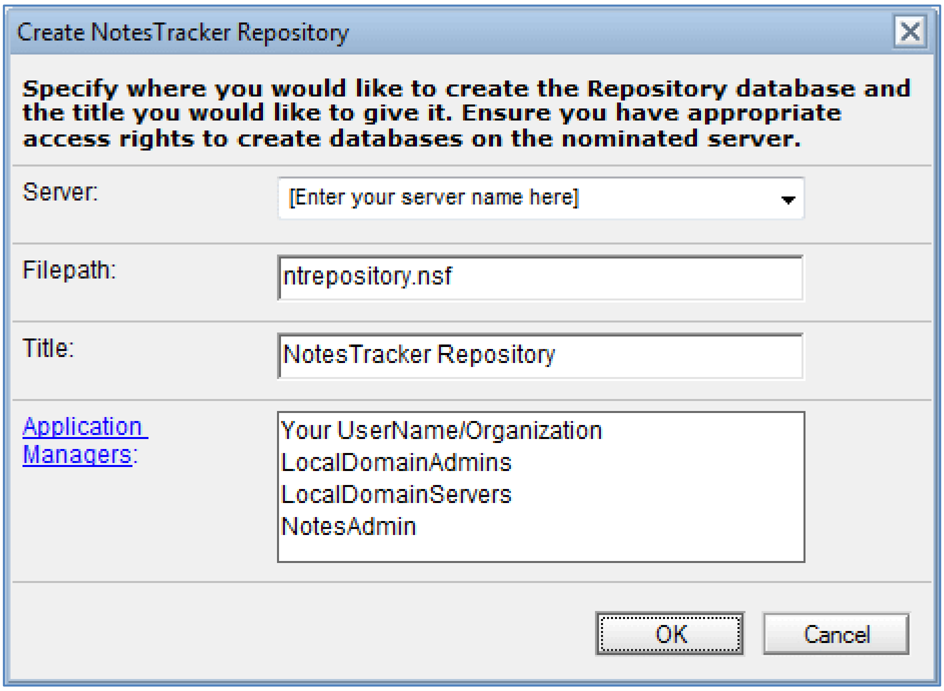
CREATING A STANDALONE REPOSITORY DATABASE

To create a NotesTracker Repository database to receive log entries from your NotesTracker enabled applications follow these steps:

- Open the NotesTracker Installer file.
- Click on the 'Create Repository' option in the navigation menu (see image below)



- Select the server (or local drive) to install the Repository database to, enter the path / filename, repository title, and select the Managers of the application, then click the Ok button.



Create NotesTracker Repository

Specify where you would like to create the Repository database and the title you would like to give it. Ensure you have appropriate access rights to create databases on the nominated server.

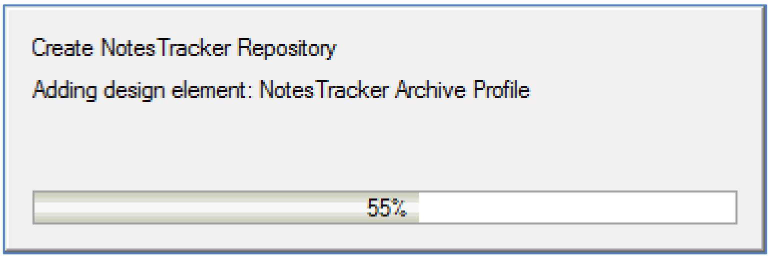
Server: [Enter your server name here]

Filepath: ntrepository.nsf

Title: NotesTracker Repository

Application Managers: Your UserName/Organization
LocalDomainAdmins
LocalDomainServers
NotesAdmin

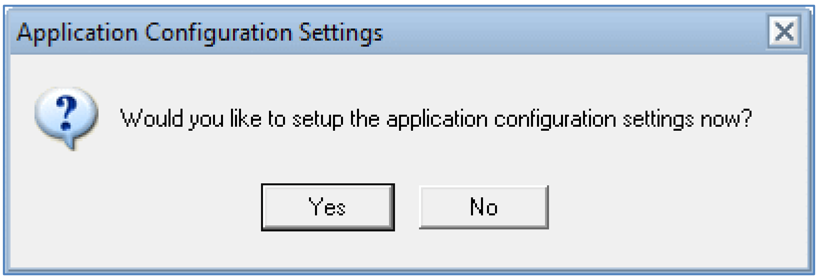
- A progress bar will display on the screen showing the various creation steps as they are performed.



Create NotesTracker Repository

Adding design element: NotesTracker Archive Profile

- When the creation of the Repository database has completed you will be provided with an option to setup the Configuration Settings for database:



Application Configuration Settings

Would you like to setup the application configuration settings now?

Click ‘Yes’ to open the Configuration Settings page.

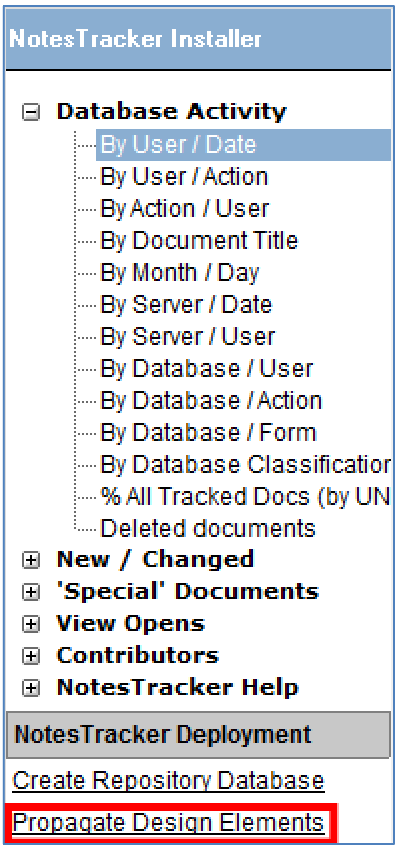
- The ‘About’ page of the Repository database will also be displayed. You can close the About page to go to the standard database screen.

PROPAGATING NOTESTRACKER DESIGN ELEMENTS TO YOUR APPLICATIONS

To add NotesTracker design elements to your own applications, or to upgrade NotesTracker design elements in existing applications to the latest release please follow the steps below.

You should be aware that if you are upgrading an application which already contains NotesTracker design elements and you have made customizations to those elements you will need to re-apply the customizations after the upgrade.

Open the NotesTracker Installer file and select the ‘Propagate Design Elements’ option in the navigation menu (see image below)



The NotesTracker Propagation Page will be displayed.

The following information should be completed.

DESIGN ELEMENTS

Copy NotesTracker configuration document

Indicates if the NotesTracker configuration document in the current database should be copied across to the target database.

Copy NotesTracker design elements

Indicates if the NotesTracker design elements should be copied to the target database.

Design Elements to Copy

Select the design elements to be copied to the target database.

- Copy basic NotesTracker elements

  This selection will copy only those design elements required for implementing tracking in the target database. All NotesTracker logs will be saved to a separate repository database.

- Copy all NotesTracker elements

  This selection will copy all NotesTracker design elements and views to the target database. All NotesTracker logs will be saved to the target database.

TARGET DATABASE

Click the ‘*Target Database*’ tab to specify which database or template the NotesTracker design elements should be added to.

Standalone Repository

Select the checkbox for this field if the target database being updated is a stand-alone Repository database (ie. the database where all NotesTracker logs are stored). Selecting this option will ensure the target database is correctly flagged as a Repository database and contains the required design elements.

Open Database

Indicates if the target database should be opened after the propagation process has completed.

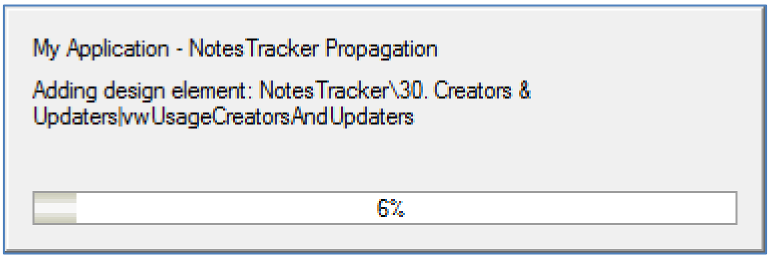
NOTES ID

You may optionally click the ‘*Notes ID*’ tab to specify an alternative Lotus Notes ID to perform the propagation with instead of the Notes ID currently being used in the Lotus Notes client.

Click the ‘Select alternate id’ link to select the ID file. A dialog box will open enabling you to select the Notes ID and provide the password for the ID file. The propagation process will automatically switch to that Notes ID when starting and then switch back to the current ID when finished.

When all information has been provided click the ‘Propagate Design’ button near the upper right-hand corner of the page. Some validation checks are performed before the propagation process begins and you’ll be advised of any problems found.

A progress bar will display on the screen showing the various propagation steps as they are performed.

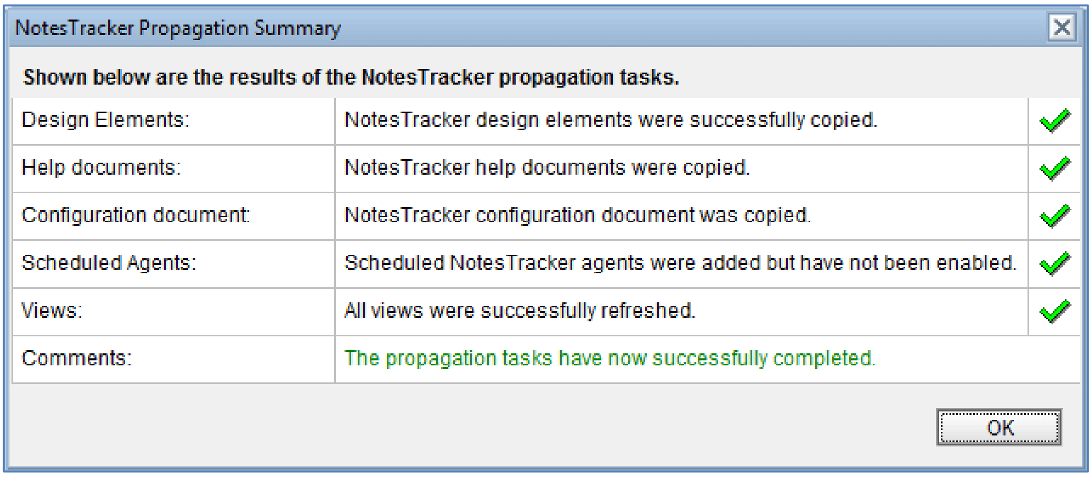


My Application - NotesTracker Propagation

Adding design element: NotesTracker\30. Creators & Updaters|vwUsageCreatorsAndUpdaters

6%

When the propagation has completed a confirmation box will display indicating the success or failure of each propagation task:

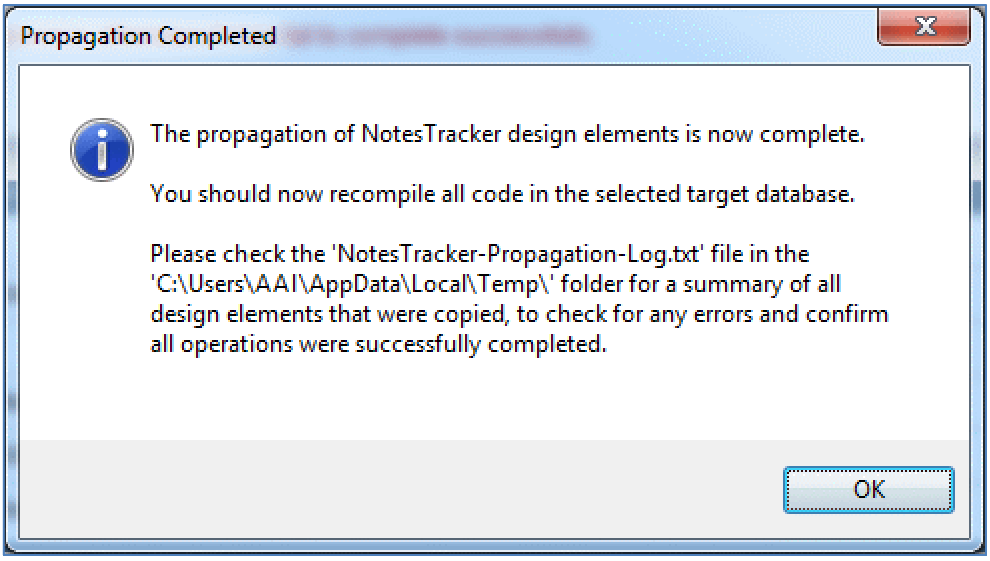


NotesTracker Propagation Summary

Shown below are the results of the NotesTracker propagation tasks.

Design Elements:	NotesTracker design elements were successfully copied.	✓
Help documents:	NotesTracker help documents were copied.	✓
Configuration document:	NotesTracker configuration document was copied.	✓
Scheduled Agents:	Scheduled NotesTracker agents were added but have not been enabled.	✓
Views:	All views were successfully refreshed.	✓
Comments:	The propagation tasks have now successfully completed.	

A log file called ‘NotesTracker-Propagation-Log.txt’ is created in the Temp folder on your machine and contains a detailed summary of the selections you made, the propagation steps performed, design elements copied, any errors that were encountered, and other general information. The path is this log file is displayed as the final step of the propagation process:



Propagation Completed

The propagation of NotesTracker design elements is now complete.

You should now recompile all code in the selected target database.

Please check the 'NotesTracker-Propagation-Log.txt' file in the 'C:\Users\AAI\AppData\Local\Temp\' folder for a summary of all design elements that were copied, to check for any errors and confirm all operations were successfully completed.

This completes the propagation of NotesTracker design elements to your own applications. You should now refer to the NotesTracker Developer Guide for information on customizations required in your application to enable it to use the NotesTracker functionality.

NOTESTRACKER CONFIGURATION SETTINGS

A **configuration document** (plus an associated configuration view) is added to each tracked database as part of the propagation process. By editing each individual database's NotesTracker Configuration document, the assigned NotesTracker Administrator for the database can for that particular database, at any time activate or deactivate usage tracking as a whole and for individual events within the database (Create documents, Read documents, Update them, Delete them, open a view, etc). The updated NotesTracker configuration document will, of course, be replicated across your Domino network just like any other document, and as soon as the updated document reaches each remote Notes database the new NotesTracker options will come into effect there.

For each database that has had usage tracking activated at some point of time: whenever one of the specified types of events occurs (create, read, update, delete¸ paste¸ etc) up NotesTracker writes a Usage Log document to the repository database that you designate. The usage log entries build up over days, weeks and months to give you an easy-to-understand picture of each application database's document and view usage.

Whenever you decide, a supplied agent can be run to archive old Usage Log entries for historical analysis and reporting purposes (in exactly the same easy and convenient way that you go about archiving your old Notes Mail entries).

A basic target of NotesTracker, but not the only one, is to measure database activity that is initiated by users directly interacting with your Notes databases, through either a Lotus Notes Client or a Web browser.

Because nearly all of the code for NotesTracker is written in LotusScript, your developers should be able to adapt and extend NotesTracker's usage logging capabilities to other environments coded in LotusScript, such as LotusScript agents – but not Formula Language agents, or Java agents (except perhaps in some way via LS2J) – running in the background (on a Domino server or Notes workstation).

The key NotesTracker functions are all written in LotusScript; while the remained of NotesTracker uses nothing else than regular simple design elements (forms, views, outlines, pages, framesets, agents, etc). This is important, because it enables a single NotesTracker installation package to run in any workstation or server platform that supports Notes and Domino.

The C API and C++ APIs were not used for NotesTracker, and neither was Java, because these languages are considerably more complex to write and deploy across all platforms. An additional disbarring consideration for Java is that it does not support the "front end" User Interface (UI) – only the "back end" server interface.

Even though LotusScript is used for the key NotesTracker functions, this does not mean that your developers necessarily need LotusScript skills, since in many instances it should be possible to deploy NotesTracker without making any changes at all to the provided code. And even if you do want to make some changes, it is likely that a lot of them will require only elementary or modest LotusScript skills.

USAGE TRACKING AND REPORTING WITH NOTESTRACKER

The distributed NotesTracker Repository database acts as:

- The **NotesTracker Toolkit** repository (or SDK – Software Development Kit), a container for NotesTracker design elements – forms, views, subforms, agents, etc – that enable you to enhance the designs your Notes application databases to incorporate NotesTracker functionality.

- An optional **repository for NotesTracker entries (or "usage log" entries)** that are generated during the logging of database-related activities when users directly interact with one or more of your Notes/Domino applications.

Usage tracking can be implemented for any Notes database, as long as you have access to its design. Naturally this means the database designs cannot be hidden. Some third-party Notes applications may have their design hidden which will prevent you from implementing usage tracking for them. Only simple designer skills are needed, unless you want to make any changes to the supplied NotesTracker code in which case a greater or lesser degree of familiarity with LotusScript is required.

In practice it would be of little value to implement usage tracking in each and every one of your Notes databases, but only in those where the ability to track usage delivers an **adequate return on investment** or provides some **worthwhile operational payback** (typically, control and privacy reasons such as the monitoring of updates to critical fields and the tracking of document deletions.

USAGE TRACKING FOR THE LOTUS NOTES CLIENT

Usage Log documents can be added to the NotesTracker database every time that one of the following events occurs in any of your tracked databases:

- ✓ **A DOCUMENT IS READ** (opened in Read mode – but never changed to edit mode and saved).

- ✓ **A DOCUMENT IS UPDATED** (opened in edit mode or opened in Read mode then changed to Edit mode, then saved).

  In most cases you will want to know more than merely that a given document was updated. NotesTracker offers **a generic, all-purpose FIELD AUDIT TRAIL capability**. When you switch on tracking of Updates, you can then also optionally switch on the tracking of **CHANGES TO THE CONTENTS OF THE DATA FIELDS IN DOCUMENTS**. When this option is activated for a database, the field "**before images**" (field contents before the Update) and "**after images**" (field contents after the Update) are placed into the Usage Log document for each document Update. When you open a Usage Log document, you'll be easily able to compare the before images with the after images, in both a "Side by Side" arrangement and an "Over and Under" arrangement, illustrated later in this guide under the heading "Tracking Changed Field Contents".

  As a refinement, you can opt to **log all fields** (both changed and unchanged) or to **log only the changed fields**. Since there's little use in logging the unchanged fields, the latter option eliminates "clutter" in the Usage Log and makes it far easier to hone in on the changed fields. This is especially true when there are many fields in a document, such as in Server Documents stored in the Domino Directory (Public Address Book) database.

- ✓ **A DOCUMENT IS CREATED** (or "composed" to use Lotus Notes terminology) – that is, created as a new document and then saved for the very first time.

- ✓ **A DOCUMENT IS DELETED** (removed permanently from the database), or **A DELETION REQUEST IS MADE** (which may fail for various reasons, typically if the user does not possess deletion rights to that database). A single document or a set of documents may be deleted in a single operation. NotesTracker will log the result for each individual document.

- ✓ **A VIEW IS OPENED** (in the Lotus Notes client) – intended to be used for once-off or occasional analysis of view usage.

- ✓ **A DATABASE IS OPENED**

- ✓ **A DOCUMENT IS PASTED** - a single document or a set of documents may be pasted in a single operation. NotesTracker will log the result for each individual document.

- ✓ **A DOCUMENT IS ADDED VIA MAIL-IN**

- ✓ **A "GENERIC" EVENT OCCURS** - with very simple design changes, your Notes developer may enable the logging of all sorts of other important application events. This extension to NotesTracker's capabilities was added to expand usage tracking to significant application events other than those listed above. Just one example of such an event would be the clicking of the "Send" button in a Mail Memo form to initiate the sending of a Notes Mail message. With a greater or lesser amount of coding almost any sort of application activity could be tracked. **This capability turns NotesTracker into a general-purpose application tracking and auditing tool.**

ABOUT DATABASE ACCESSES VIA A NOTES CLIENT

When you create or edit a document via a Lotus Notes Client, a usage log entry is written only once – when the document is closed – regardless of how many times the document is saved during the editing process.

This is an intentional feature of NotesTracker, designed to report only the **net result** of the editing process. (If you really wanted to, it would be a simple matter for you to alter the NotesTracker routines to capture what is changed for each and every Save.) Another major benefit of this approach is the significant reduction in the number of Usage Log entries written. This not only conserves disk space (plus processor cycles and network traffic) but also facilitates your metrics analysis by eliminating the clutter that logging of multiple Saves would cause.

The field “before images” (field contents prior to change) are truly those that the user saw when she/he opened the document for editing.

USAGE TRACKING FOR WEB BROWSERS

With a few simple additions to a database's design, NotesTracker will track the creation, updating and reading of Notes documents via a web browser. The actions are recorded in same general format as in the Usage Log repository in the same fashion as was done for the Lotus Notes Client in previous versions of NotesTracker.

This provides you with a different, more incisive yet simple way to track and analyze your Domino (web based) document activity than is provided by some other Domino web tracking products. These others all rely on what is written to the DomLog.NSF database for their statistics, and they can’t offer the same sort of detail that NotesTracker does – such as comparing before/after contents of all fields in a document, to name one. Furthermore, the conventional Web Logs are usually overloaded with a maze of trivial, uninteresting information (such as the names of the myriad insignificant image files which form part of a typical Web page).

NotesTracker has a more **application-centric** approach, cantering on “CRUD” – document Creates, Reads, Updates (including field content changes), and Deletes.

ABOUT DATABASE ACCESSES VIA A WEB BROWSER

The nature of the HTTP protocol is for web pages to be sent out by the HTTP server (Lotus Domino, or any other), via a POST operation. This is “set and forget” or "stateless" style of operation. The server may receive the page back from the browser within a second or two of the POST, within some short or long period of minutes or hours, or may never receive the page back at all. Only when the user clicks a SUBMIT button in the browser page does the web server (via a GET operation) obtain incoming field contents. The HTTP protocol has no mechanism that automatically relates the fields in the page that was sent out by the server to the fields in the page that was just returned to the server. This means that there’s no easy way to compare the page’s field contents before and after they are updated.

At the time that the browser page is submitted back to the Domino server, a copy of the so-called “back-end” document is retrieved from the database on the Domino server and the fields from this freshly-retrieved document are used as the “before images”.

Note: It is important to be aware that the logged contents of such fields may possibly not be the same as the contents of the fields that initially were sent out to the browser page. There is always the chance – perhaps only slight – that some other user(s) might have updated the back-end document in the period between the POST and GET operations. It’s hard to come up with a foolproof solution for this issue, which in essence is caused by the “statelessness” of web browser sessions. (In terms of field content changes, this is analogous to the generation of Save Conflicts, caused when multiple users simultaneously update a document via a Notes Client.)

THE NEED TO 'SIGN' THE NOTESTRACKER WEB AGENTS

Refer to the Administration Topics section below for more details, but it should be pointed out at this early stage that, for security reasons, in the Domino server environment web agents need to be appropriately “signed”. If the NotesTracker web agents are not signed so as to be acceptable for your Domino server then they will fail to execute, which means that NotesTracker will not log any Web browser interactions. (This is a normal Domino security consideration and not a NotesTracker limitation.)

TRACKING ACTIONS CARRIED OUT OTHER THAN VIA A NOTES CLIENT OR A WEB BROWSER

NotesTracker was conceived primarily to track actions performed by a real person acting directly on Notes documents via the so-called “front end” -- the graphical user interface or “UI” provided by a Lotus Notes Client (or perhaps a Web browser).

NotesTracker also tracks some types of actions that are **database-level events** rather than document-editing type of events:

- **Document Deletions** - commonly initiated at the front end, but can be carried out by back-end processes, and normally are not executed on a document that has been opened. They are recorded as database-level events.

- **Document Pastes** - carried out at the front end, but not involved with the opening of a document (followed by its closing). They also are recorded as a database-level event.

- **Document Mail-ins** - operate in the back end, adding one document at a time to the database. These too are recorded as a database-level event, but are quite unlikely ever to seriously affect performance.

Note: a single deletion or paste request may involve either a single document or multiple documents, and can be performance intensive since NotesTracker will write one Usage Log document per document deleted or pasted. Be aware that both deletes and pastes are **requests** that might fail (if the user has insufficient rights to the database). All the same, NotesTracker will faithfully log each such request even if it fails. A paste operation occurs synchronously and its result is immediately recordable. On the other hand, we have discovered that deletions occur asynchronously – even though they might *appear* to be synchronous – and that the result of a deletion request is impossible to log with absolute certainty. For this reason you will see some deletions logged as “indeterminate” (rather than successful or failed). This seems the best that NotesTracker can accomplish, since it operates at the LotusScript event level. Nevertheless, NotesTracker’s logging of deletion events is till useful.

For types of events other than all the above, the fact that NotesTracker routines are structured in a modular fashion could make it easy for you to adapt them for tracking other types of actions, such as button clicks or agents running in the background, on the Notes Client or Domino server. This might be important for you to do in some databases, for completeness of usage metrics where you have the need to track other classes of events (or document actions even when they are not performed via the front end).

You could adapt the NotesTracker code to run in any agents written in LotusScript, since the NotesTracker routines were developed in this language. (LotusScript was chosen for NotesTracker since this language has all the features needed to perform the fairly complex tasks involved in usage tracking, in both the front end and the back end. The Notes Formula Language does not have the programmability, and the Java language only works in the back end.

WHAT TO TRACK? INDIVIDUAL NOTES DATABASES OR SETS OF DATABASES?

You have considerable flexibility in the way you deploy NotesTracker to gather application usage metrics about your various databases. NotesTracker can be used to track activity in individual Notes databases, or in related sets of Notes databases – as few or as many as you desire.

One user even justified purchasing a corporate NotesTracker license purely to track changes being made to a single database (which happened to be the Domino Directory, a.k.a. the Public Address Book).

You do not have to track activity in each and every database, and the degree of tracking can vary from database to database. Indeed, you will probably only wish to gather usage metrics for a limited number of databases – or even just a single database – where you see definite value and payback.

Furthermore, you can easily vary the nature or degree of tracking in a given database if your tracking requirements for the database vary over time. For example, you may want always to track document Deletions and Creates/Updates in a certain business-critical database, but only track document Reads in that database for periods of a day or two every now and then (just to get a feel for how overall use of the database is trending).

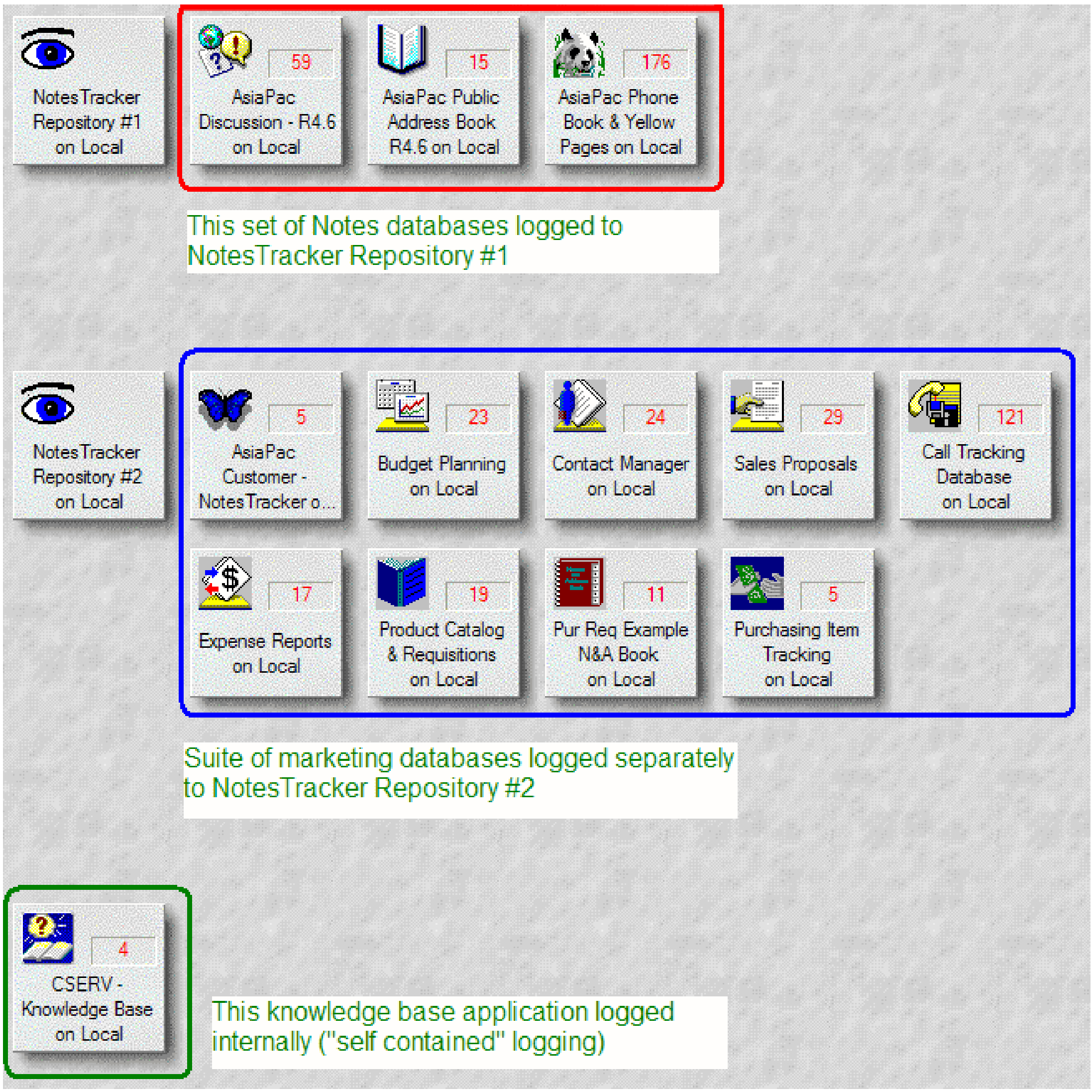
How is this achieved? As fully explained later in this guide, each database that you wish to track via NotesTracker must have its own **NotesTracker Configuration Document**. It is very easy, without any developer intervention needed, to edit the various control values in this document thereby changing the nature and extent of usage tracking (for that individual database) from that point of time onwards. And, as soon as the updated configuration document is replicated across your Domino server and Notes Client network, the updated logging control options take universal effect.

What is more, it is a simple matter to have a given database tracked in isolation or as part of a related set of databases. This is determined merely by a setting in each database's NotesTracker Configuration Document which controls where the usage log entries are written (either to an isolated NotesTracker Repository or to a shared NotesTracker Repository). The decision is entirely yours, and you can easily adjust the various control settings as and when your usage tracking requirements change.

If it makes sense to track a database as part of a set (or “suite”) of related databases, you merely specify that a single NotesTracker Database is to act as the repository of the usage log documents for all of the databases in that set. Different sets of Notes databases can have their usage log documents written to different NotesTracker Repository databases. This gives you great flexibility in configuring database usage logging for your Notes/Domino applications, and in the metrics analyses that you perform subsequently.

It does not end there. If it makes more sense for whatever reason to track a particular database in isolation, then you can specify a NotesTracker Database that is to act as a repository of the usage log documents just for that single Notes database. Going one step further, there is even an option to specify that "**internal tracking**" should occur, in which case the NotesTracker usage log documents are written to the database itself (rather than to an external NotesTracker Repository database). This enables usage tracking to be "**self-contained**" within that particular Notes database – the database acts as its own repository of usage log entries – and this may be quite desirable in some situations.

The following diagram illustrates this quite clearly:


NotesTracker Repository #1 on Local
59
AsiaPac Discussion - R4.6 on Local
15
AsiaPac Public Address Book R4.6 on Local
176
AsiaPac Phone Book & Yellow Pages on Local
NotesTracker Repository #2 on Local
5
AsiaPac Customer - NotesTracker o...
23
Budget Planning on Local
24
Contact Manager on Local
29
Sales Proposals on Local
121
Call Tracking Database on Local
17
Expense Reports on Local
19
Product Catalog & Requisitions on Local
11
Pur Req Example N&A Book on Local
5
Purchasing Item Tracking on Local
4
CSERV - Knowledge Base on Local

This set of Notes databases logged to NotesTracker Repository #1

Suite of marketing databases logged separately to NotesTracker Repository #2

This knowledge base application logged internally ("self contained" logging)

Detailed NotesTracker set-up considerations, together with the advantages and disadvantages of the various logging options, are discussed in depth later in this guide.

In summary, NotesTracker affords you unique flexibility in deciding how databases are grouped for usage tracking. What is more, the arrangement is quite easily modified as your grouping requirements change. You simply change settings in the NotesTracker Configurations document in the affected database(s), and then just wait a little while for the NotesTracker Configuration document(s) to replicate around your Domino network for the changes to take effect globally.

Note: an available feature is the ability to specify a database's application "classifiers." You would classify a database using terms such as Marketing, Support, Administration, Finance, Human Resources, CRM, or whatever is relevant. You can specify one or more classifiers per database. There is an associated new view in the NotesTracker Repository that enables you to **examine all of the actions performed against your databases categorized by classification of application**. See the '*Specify a Database's Application Classification*' topic in this guide for more details about database classifiers.

USING NOTESTRACKER WITH CLUSTERED DOMINO SERVERS

You should be aware that, as distributed, NotesTracker uses simple database file open operations, rather than “Open with Failover”. In the event of a failover, it is reasonably likely that the use of a specific server name and file path will cause the database open to fail. Therefore, in a clustered Domino server environment, opening by Replica ID would be the preferred option.

Otherwise, your Notes developer could decide to modify the distributed NotesTracker code so as to use the “Open with Failover” method to handle the clustered server situation. Since there is only a single field in the NotesTracker Configuration Document for the Path and Filename, you would have to be careful to deploy the database using the same Path and Filename on each server. (It would be possible to modify the NotesTracker Configuration Document’s form design and tracking routines to cater for different paths and file names on individual clustered servers, but probably not worth the effort – not to mention that the increased complexity could cause administration and/or operational problems.)

PLACEMENT OF REPLICAS OF THE NOTESTRACKER DATABASE

For best response times, it would probably be best to place the NotesTracker Database's replica on each Domino server.

You should also consider having a replica of the NotesTracker Database on a user's Notes Client workstation or notebook computer if they are working with NotesTracker enabled applications locally. By doing this, usage can be tracked even for users without "live" connection to the Domino server network.

If the Notes Administrator (or perhaps the Notes "power user") sets up one-way selective replication from the local NotesTracker replica to the server-based replica, this will keep down both the local database size and replication traffic over the network, while enabling Usage Log documents to be sent in even from "mobile" users, so as ultimately to be centrally consolidated and analyzed for an overall analysis of usage.

HOW TO SPECIFY THE NOTESTRACKER TARGET REPOSITORY FOR A DATABASE BEING TRACKED

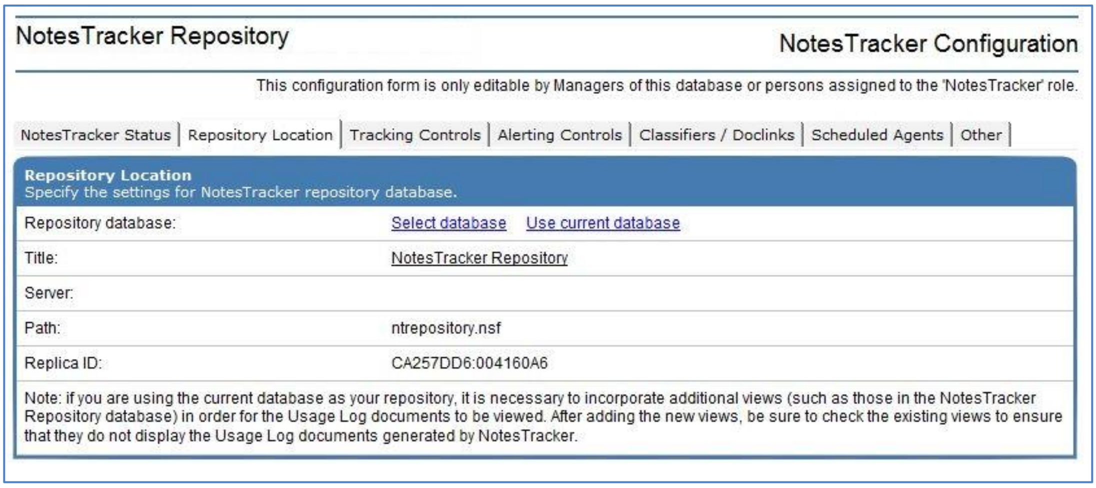
The usage tracking code is added to Notes databases and is activated by events such as opening Notes documents or views, deleting documents, and saving documents with new fields or updated field contents.

For each such event, the code writes (appends) a new Notes document that we call a “Usage Log record” into a NotesTracker target database or “repository”. This means of course that NotesTracker must have a means of specifying, for each database being tracked, the repository to which such Usage Log documents are to be written.

The NotesTracker administrator and/or Notes Administrator will be involved in determining where on the Domino server (or Notes Client workstation, in the case of local usage tracking) each NotesTracker Repository is located.

A quite simple mechanism is used for this. You must set up in advance, in each Notes database being tracked, a **NotesTracker Configuration Document** that specifies the name and location of the NotesTracker Repository for that particular database. If the NotesTracker Repository is subsequently relocated, naturally the NotesTracker Configuration Document will have to be correspondingly updated to reflect the new repository location.

To specify the target repository open the NotesTracker Configuration Settings document and select the ‘Repository Location’ tab:



NotesTracker Repository

NotesTracker Configuration

This configuration form is only editable by Managers of this database or persons assigned to the 'NotesTracker' role.

NotesTracker Status | Repository Location | Tracking Controls | Alerting Controls | Classifiers / Doclinks | Scheduled Agents | Other

Repository Location
Specify the settings for NotesTracker repository database.

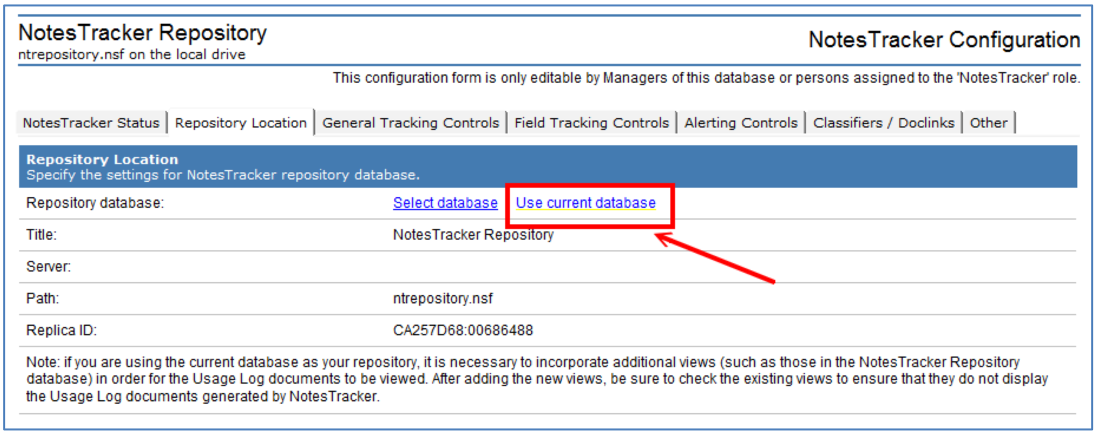
Repository database:	Select database Use current database
Title:	NotesTracker Repository
Server:	
Path:	ntrepository.nsf
Replica ID:	CA257DD6:004160A6

Note: if you are using the current database as your repository, it is necessary to incorporate additional views (such as those in the NotesTracker Repository database) in order for the Usage Log documents to be viewed. After adding the new views, be sure to check the existing views to ensure that they do not display the Usage Log documents generated by NotesTracker.

The repository location may be the current database or a separate Repository database.

INTERNAL (OR "SELF CONTAINED") LOGGING TO THE CURRENT DATABASE

For a particular Notes database, instead of directing Usage Log entries to an external NotesTracker repository it might be better to quickly modify that database's design so that if you specify that logging is internal you will be able to view the log entries that NotesTracker writes to the database. This makes usage tracking "self- contained" to that database, and in some cases this is very convenient (See, for example, the SDMS and CAPTURE free database applications that are downloadable from https://www.agecom.com.au/NotesTracker)



NotesTracker Repository
ntrepository.nsf on the local drive

NotesTracker Configuration

This configuration form is only editable by Managers of this database or persons assigned to the 'NotesTracker' role.

NotesTracker Status | Repository Location | General Tracking Controls | Field Tracking Controls | Alerting Controls | Classifiers / Doclinks | Other

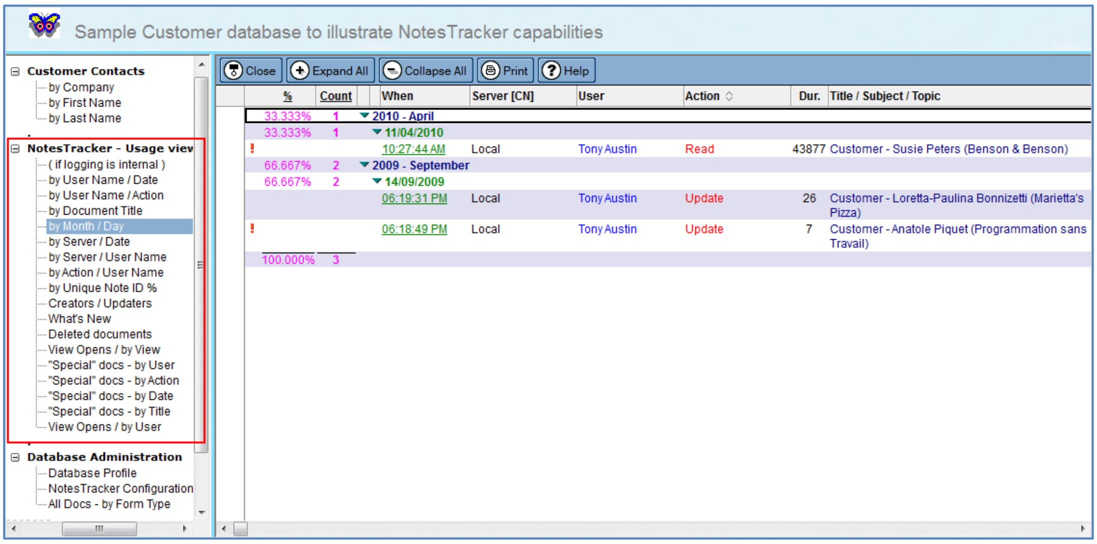
Repository Location
Specify the settings for NotesTracker repository database.

Repository database:	Select database Use current database
Title:	NotesTracker Repository
Server:	
Path:	ntrepository.nsf
Replica ID:	CA257D68:00686488

Note: if you are using the current database as your repository, it is necessary to incorporate additional views (such as those in the NotesTracker Repository database) in order for the Usage Log documents to be viewed. After adding the new views, be sure to check the existing views to ensure that they do not display the Usage Log documents generated by NotesTracker.

Modifying the design will be quite easy. Essentially, it will be just a matter of adding a few Usage Log views (maybe even just a single view) and updating the database's navigator to include the added views.

Here's an example of a database with Usage Log views added and navigator updated (as highlighted in red). It's the **NotesTracker Customer DB** that is one of the sample NotesTracker enabled applications available for download from the AGECOM website:



Sample Customer database to illustrate NotesTracker capabilities

	%	Count	When	Server [CN]	User	Action	Dur.	Title / Subject / Topic
	33.333%	1	2010 - April					
	33.333%	1	11/04/2010					
!			10:27:44 AM	Local	Tony Austin	Read	43877	Customer - Susie Peters (Benson & Benson)
	66.667%	2	2009 - September					
	66.667%	2	14/09/2009					
			06:19:31 PM	Local	Tony Austin	Update	26	Customer - Loretta-Paulina Bonnizetti (Marietta's Pizza)
!			06:18:49 PM	Local	Tony Austin	Update	7	Customer - Anatole Piquet (Programmation sans Travail)
	100.000%	3						

One advantage is that if you select this option, the Usage Log documents may be written more rapidly than if they are directed to an external database.

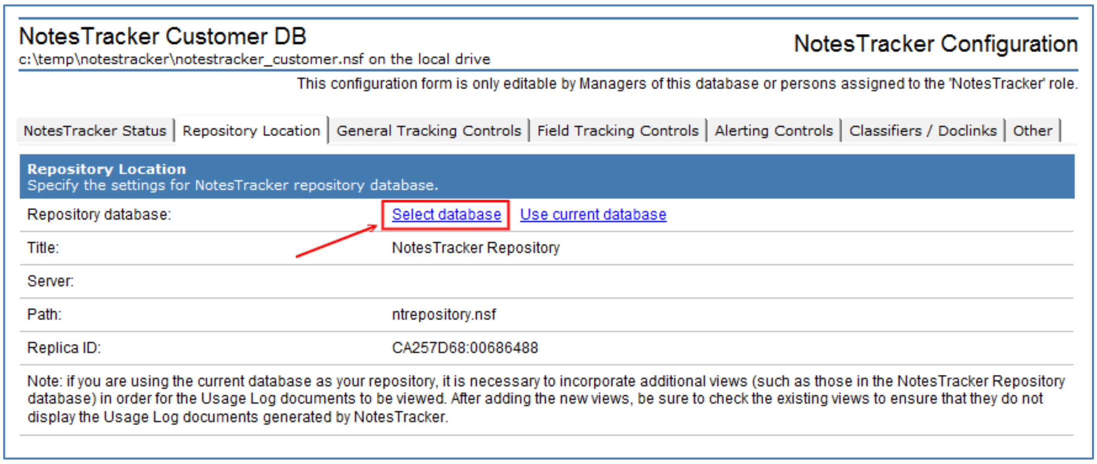
On the other hand, the current database's size will grow – perhaps rapidly – due to the additional documents being added. Also, the view indices will be built for the views you add to display the usage metrics. The impact on extra disk space occupied by the database and extra processor cycles must be taken into account when managing Domino servers and Notes Client workstations that hold the database.

USING MULTIPLE NOTESTRACKER REPOSITORIES, FOR DIFFERENT SETS OF APPLICATIONS OR DIFFERENT PERIODS

If you wish to track different groups or suites of Notes database applications separately, this is quite simple to do!

All that it's necessary to do is to create a separate non-replica copy of the NotesTracker Repository Database for each such set or suite of application databases.

Then you direct the Usage Tracking (logging) activity to appropriate Repository database merely by opening the NotesTracker Configuration document and selecting the location of the Repository database.



NotesTracker Customer DB
c:\temp\notestracker\notestracker_customer.nsf on the local drive

NotesTracker Configuration

This configuration form is only editable by Managers of this database or persons assigned to the 'NotesTracker' role.

Repository Location
Specify the settings for NotesTracker repository database.

Repository database:	Select database Use current database
Title:	NotesTracker Repository
Server:	
Path:	ntrepository.nsf
Replica ID:	CA257D68:00686488

Note: if you are using the current database as your repository, it is necessary to incorporate additional views (such as those in the NotesTracker Repository database) in order for the Usage Log documents to be viewed. After adding the new views, be sure to check the existing views to ensure that they do not display the Usage Log documents generated by NotesTracker.

Similarly, you could use different replicas of the NotesTracker Repository Database to track activity in separate time periods. For example, you could switch over to a different NotesTracker Repository Database for each year, each quarter, or (if you have large amounts of activity) each month.

SECURITY, ACCESS CONTROL AND PRIVACY CONSIDERATIONS

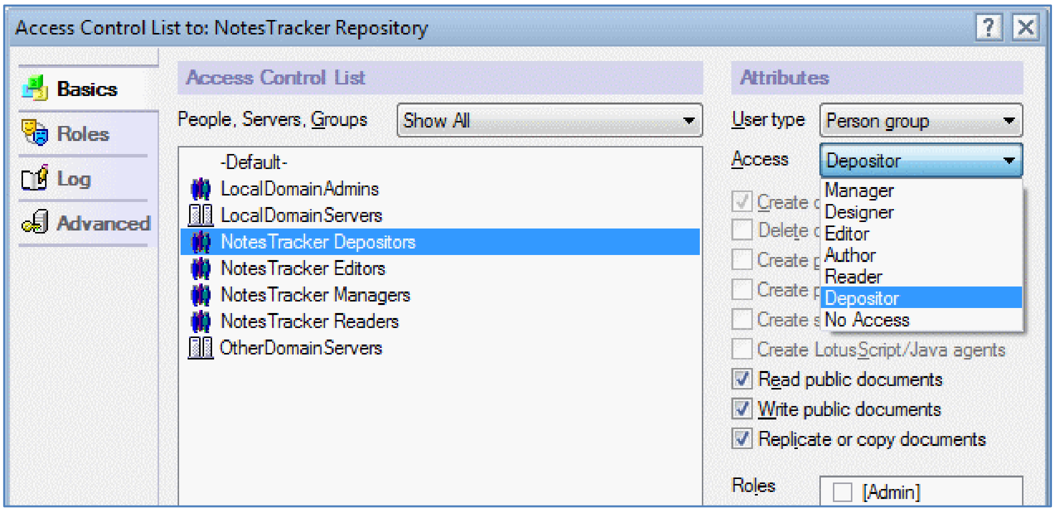
The distributed NotesTracker Repository database has the following Access Control List (ACL) settings:

- **Default access should be "Author"** (or perhaps you can try “Depositor”).

  Usage tracking will not operate unless Usage Log documents can be added to the NotesTracker Repository database. This is what happens implicitly, without the user being aware of it, for every tracked action (Create, Read, Update, Delete, Paste or Mail-in of a document, and opening of a View or Database).

  For NotesTracker logging to succeed, a basic requirement is that each user (or background agent) involved must have the authorization level needed to **create documents** into the repository. An alternative would be to give them the authorization to **deposit documents** into the database

  You might decide to create a person groups for this – say, a group named "**NotesTracker Authors**" (with Create authorization) or “**NotesTracker Depositors**” (with deposit authorization). Such groups might be used against all NotesTracker repositories, but you might need more fine-grained control and therefore set up groups specific to each particular repository.



Access Control List to: NotesTracker Repository

People, Servers, Groups: Show All

-Default-
LocalDomainAdmins
LocalDomainServers
NotesTracker Depositors
NotesTracker Editors
NotesTracker Managers
NotesTracker Readers
OtherDomainServers

User type: Person group

Access: Depositor

☑ Create d
☐ Delete d
☐ Create p
☐ Create p
☐ Create s
☐ Create LotusScript/Java agents
☑ Read public documents
☑ Write public documents
☑ Replicate or copy documents

Roles: ☐ [Admin]

Also, you might create and use these other person groups:

- "**NotesTracker Readers**" group, with Reader access – not allowed to delete documents – for people who need to be able to see the usage tracking views.

- "**NotesTracker Editors**" group, with Editor access – and optionally allowed to delete documents, if you want them to have the ability to "clean" the NotesTracker Database by manually deleting documents from it (there may be situations where unwanted log documents appear in the database).

- "**NotesTracker Managers**" group, with full Manager access (for the usual reasons).

IMPORTANT NOTE: only database managers or those with the [Admin] role have the ability to create and modify the NotesTracker Configuration document. This is explained in the NotesTracker Developer Guide.

Another consideration is that replica copies of your Notes databases will often be deployed not just on Domino servers but also "locally" (that is, on Notes Client Workstations and notebook computers). If a database's ACL is not set to "**Enforce a consistent Access Control List across all replicas of this database**" then it is quite possible (or even likely) that the Notes user – being by default the Manager of the locally-deployed database – may change some of the usage tracking control settings in the NotesTracker Configuration document. In the worst case, the user may even completely switch off usage tracking. Furthermore, any such change to the NotesTracker Configuration will probably replicate throughout your Domino server network and disrupt usage tracking on a wide scale!

These are only suggested ACL settings. They are NOT mandated by NotesTracker's usage tracking code, and you can set up your own alternatively-named groups or use existing ones with whatever access rights you deem appropriate.

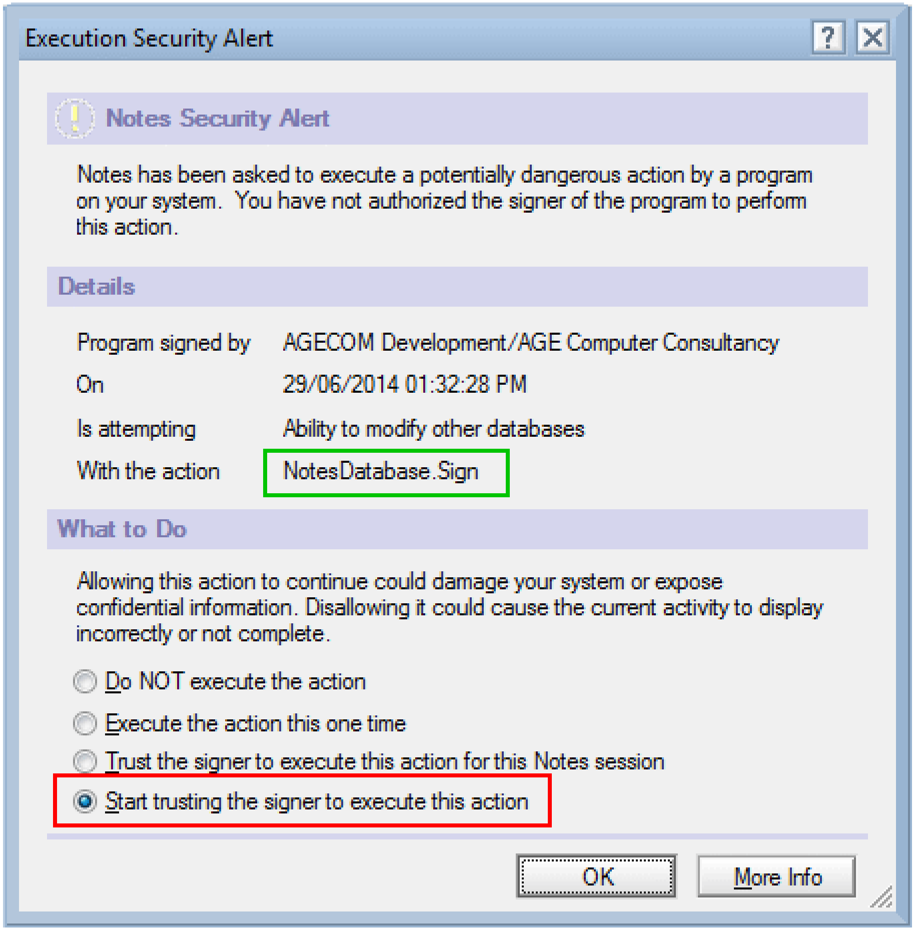
AUTHORIZING THE NOTESTRACKER AGENTS BY SIGNING THEM

If you want to track document usage activity against a database that is performed via a Web browser, then (as described in the NotesTracker Developer Guide) it is necessary for two Web agents to be included in the database. The two agents are **NotesTrackerWebQueryOpen** which tracks documents being opened by the browser, and **NotesTrackerWebQuerySave** which tracks documents being created and fields being updated. (The same authorization considerations may apply to other agents used by NotesTracker, such as the **Paste** and **Mail-In** agents, even though these do not operate in a Web environment.)

It is beyond the scope of this document to go into details about the need to **sign** an agent so that it has sufficient rights to execute on a domino server. Please refer to the Lotus Domino Administrator Help for an explanation of the procedure for signing agents, as well as how to decide which ID file has appropriate execution rights and so can successfully be used to carry out the signing.

To save you the trouble of going through the signing process manually – via the Lotus Domino Administrator interface – there is an agent that will sign these two agents automatically. Simply log on to Lotus Notes (not the Domino Administrator) **with the appropriately authorized ID file**, and execute the agent named **Sign the NotesTracker web agents** (which will sign just the two above- named Web agents and not any other design elements in the database).

You will probably get a dialog box like the following:



Execution Security Alert

Notes Security Alert

Notes has been asked to execute a potentially dangerous action by a program on your system. You have not authorized the signer of the program to perform this action.

Details

Program signed by AGECOM Development/AGE Computer Consultancy

On 29/06/2014 01:32:28 PM

Is attempting Ability to modify other databases

With the action NotesDatabase.Sign

What to Do

Allowing this action to continue could damage your system or expose confidential information. Disallowing it could cause the current activity to display incorrectly or not complete.

○ Do NOT execute the action

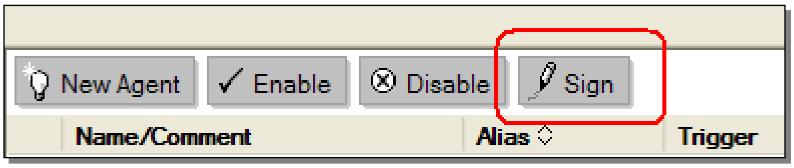
○ Execute the action this one time

○ Trust the signer to execute this action for this Notes session

● Start trusting the signer to execute this action

Select "***Start trusting the signer to execute this action***" and click the **OK** button.

Alternatively, if you are using Domino Designer 6 or later you can sign each agent using the button in the Agent List design view, as follows:

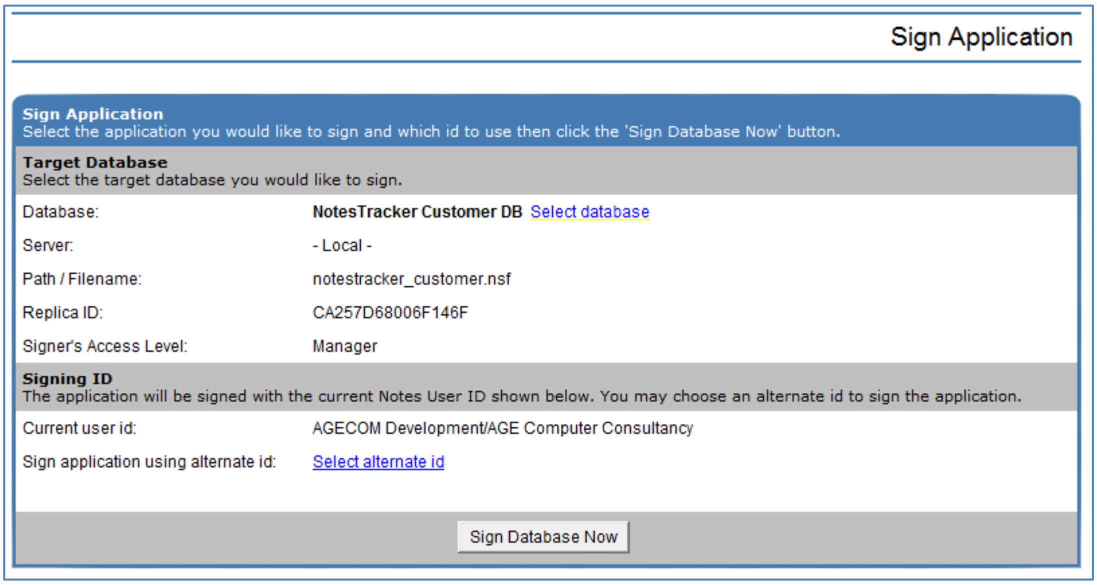


Note: you must sign the agents with an **appropriately authorized ID file** (an ID with sufficient authority to allow the two agents to run on the Domino server).

THE “SIMPLE SIGNER” DATABASE SIGNING TOOL (FREE)

An alternative solution freely available from AGECOM is an easy-to-use utility designed purely for signing Notes databases. You can download this free "**Simple Signer**" tool from the AGECOM website at: https://www.agecom.com.au/SimpleSigner

It is a Notes application that you merely open and click the ‘Select Database to Sign’ button at the top of the view. The following page is displayed enabling you to select the target database to sign and the Notes ID to use for signing the database with. You then only have to click the ‘Sign Database Now’ button to carry out the signing process. In action, the Simple Signer will look something like this:



Sign Application

Sign Application
Select the application you would like to sign and which id to use then click the 'Sign Database Now' button.

Target Database
Select the target database you would like to sign.

Database: **NotesTracker Customer DB** Select database

Server: - Local -

Path / Filename: notestracker_customer.nsf

Replica ID: CA257D68006F146F

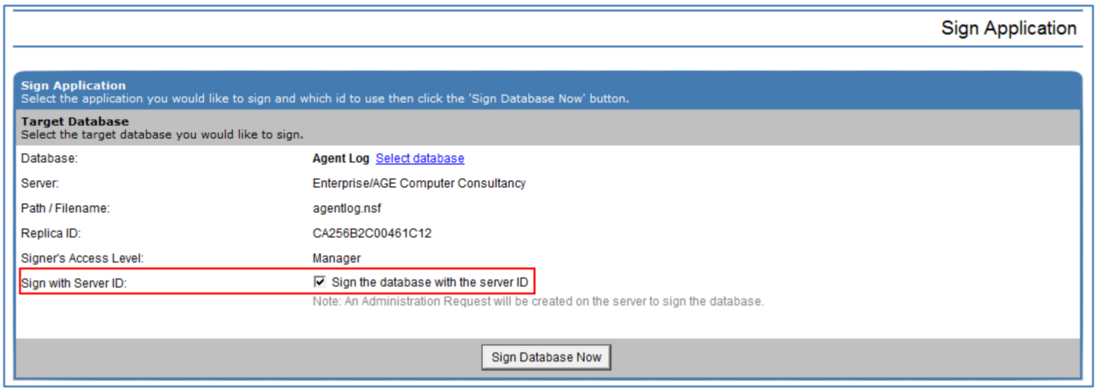
Signer's Access Level: Manager

Signing ID
The application will be signed with the current Notes User ID shown below. You may choose an alternate id to sign the application.

Current user id: AGECOM Development/AGE Computer Consultancy

Sign application using alternate id: Select alternate id

For server-based databases an option is available to sign the target application with the server’s id.



Sign Application

Sign Application
Select the application you would like to sign and which id to use then click the 'Sign Database Now' button.

Target Database
Select the target database you would like to sign.

Database: **Agent Log** Select database

Server: Enterprise/AGE Computer Consultancy

Path / Filename: agentlog.nsf

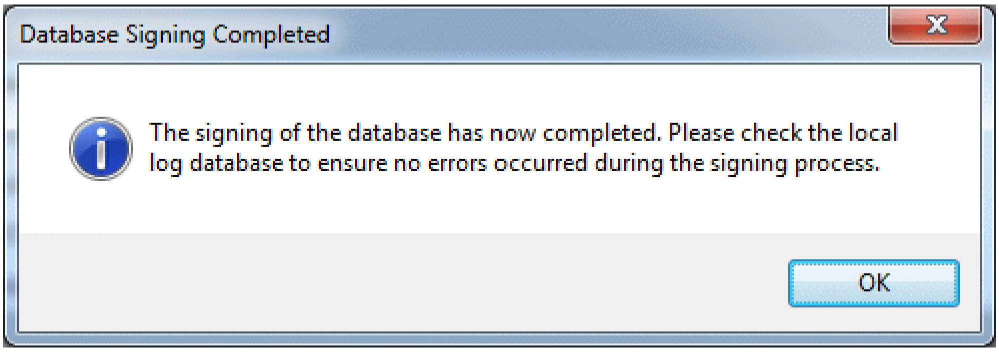
Replica ID: CA256B2C00461C12

Signer's Access Level: Manager

Sign with Server ID: ☑ Sign the database with the server ID
Note: An Administration Request will be created on the server to sign the database.

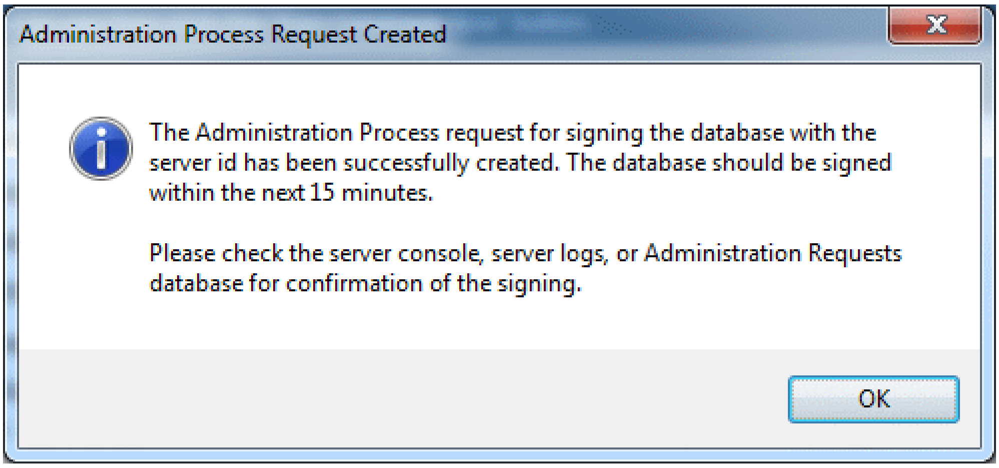
Selecting this option will create an Administration Request on the server to sign the target database.

After the ‘Sign Database Now’ button has been clicked the following message will be displayed if the signing was successful:


Database Signing Completed

The signing of the database has now completed. Please check the local log database to ensure no errors occurred during the signing process.

For a server-based database if you’ve chosen to sign the database using the server the following message will be displayed if the Administration Request to sign the database was successfully submitted:


Administration Process Request Created

The Administration Process request for signing the database with the server id has been successfully created. The database should be signed within the next 15 minutes.

Please check the server console, server logs, or Administration Requests database for confirmation of the signing.

If any errors occur while attempting to sign the database or submit the Administration Request you will receive the appropriate error notification.

A log record is written to the Simple Signer application for each signing request. This enables you to quickly see which applications have been signed, when they were signed, and which Notes ID was used.

More information about using the Simple Signer application can be found in its “Help Using This Database” document.

SECURITY AND PRIVACY CONSIDERATIONS FOR ADMINISTRATORS AND DEVELOPERS

⚠

IMPORTANT

NotesTracker Security & Privacy Considerations for Administrators / Developers

When implementing usage logging in your databases, **be very careful not to allow confidential or sensitive information to be logged**. Logged information might not be appropriate for general consumption.

Even the document titles may give clues to confidential, sensitive or personal information. Consider the disastrous implications of just the mere mention in a Usage Log document's title of such things as a proposed merger or acquisition, legal action, an employee's possible termination, or many other such matters!

You should thoroughly test the logging activity before deployment to ensure that confidentiality and privacy are maintained appropriately.

It may even be that certain databases, or at least aspects of them, should not be tracked. Developers and administrators must always keep this in mind.

SAMPLE PRIVACY NOTICE FOR TRACKED DATABASE USERS

The database contains a form which you can use as a starting point or guideline for inclusion in each tracked database as a privacy/awareness notice to your users.

PRIVACY NOTICE (SAMPLE)

Important information regarding Usage Tracking

Usage Tracking may be activated continuously or from time to time so as to record activity against the documents in this database. A snapshot is taken of information about all users (such as user name, date/time and server) plus such things as document "Read", "Create", "Update" and "Delete" operations, and field changes.

Why are we tracking usage?

Tracking Read access lets us see what parts of the database are getting the most "hits", enabling better knowledge management of the database (e.g. what is useful and should stay in, and what is rarely used and could be dropped out).

Tracking allows us to proactively manage potential opportunities or issues. For instance, we may find that a particular database or server is being heavily used, or that certain documents are noticeably more popular than others. We can see from the usage pattern how to improve the information provided via the database or perhaps that we need to improve the design of the database so that you can use it more efficiently and effectively.

Usage tracking can enable a "Latest News" or "What's New" function to be added for the database. This would identify all new or recently-updated updated documents that could (for example) be displayed on the home page and so keep you better informed.

Each time that a contributor to the database creates or updates a document, a record of the change is filed by the Usage Tracker and a list of these documents could be displayed in descending date order (from today) in "Latest News". This enables you to identify new or modified documents at a glance, rather than having to scour the containing database (or databases) for any changes. The information collected by the Usage Tracker will not be used for any purpose other than those outlined above.

NOTESTRACKER CONFIGURATION – STARTING OFF

THE NOTESTRACKER CONFIGURATION SETTINGS DOCUMENT

A **single** NotesTracker Configuration Document is needed in each database that is being tracked. It is used by each and every NotesTracker routine (whether running on a Notes Client, a Web browser, or as Domino server- based agent).

If there is no configuration document in a database, then (in order to not interfere with user access to the database) the NotesTracker routines detect this and no tracking occurs for that database until the configuration document is subsequently added.

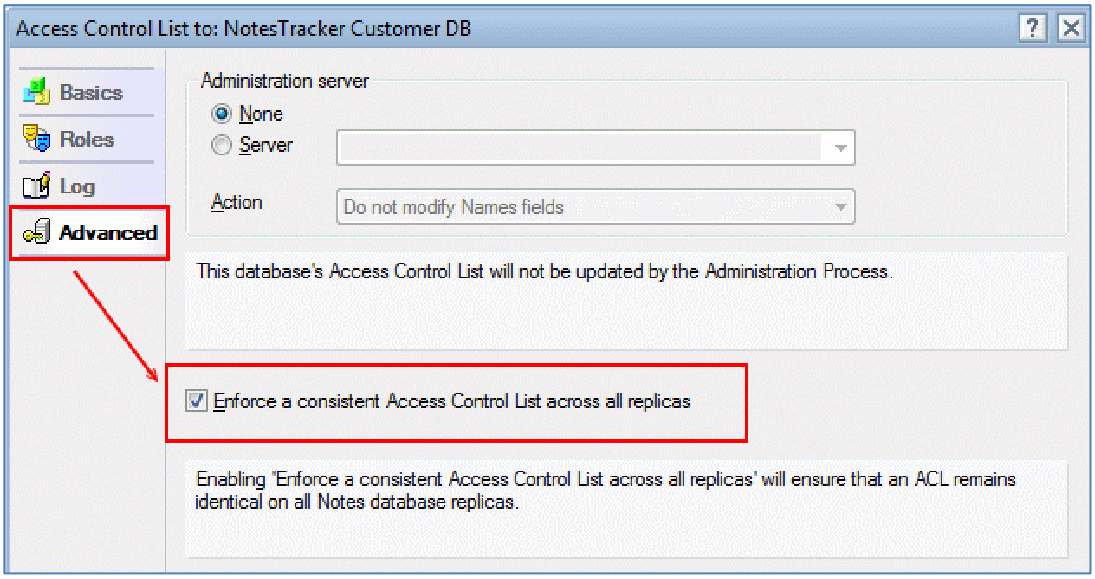
You are prevented from manually creating more than one such configuration document per database. However, experience has shown that multiple configuration documents can sometimes be present, presumably due to replication faults. It is safe (and probably wise) to delete superfluous configuration documents, leaving only one per database.

ACCESS LEVEL REQUIRED FOR EDITING THE CONFIGURATION DOCUMENT

Because the NotesTracker Configuration Document is the key means to control whether usage logging is active, and which tracking options are in effect for a database, it is most important that the ability to edit this document be restricted.

Therefore, only those with Manager level access in a database's ACL are allowed to edit the NotesTracker Configuration Document. (This is controlled by code in the QuerySave event of the NotesTracker Configuration form.)

Note: following normal Notes behavior, the checking of access level will only effective if the database is opened from a Domino server. For databases located locally on the Lotus Notes client the ACL is only respected if the 'Enforce a consistent Access Control List across all replicas' option has been enabled in the Advanced section of the ACL as per the following image:



Administration server

● None

○ Server

Action: Do not modify Names fields

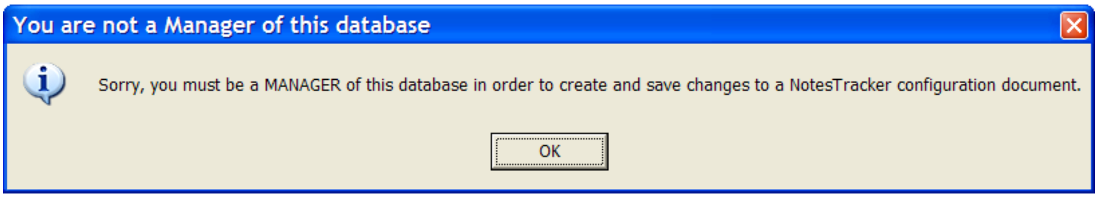
This database's Access Control List will not be updated by the Administration Process.

☑ Enforce a consistent Access Control List across all replicas

Enabling 'Enforce a consistent Access Control List across all replicas' will ensure that an ACL remains identical on all Notes database replicas.

If the database is opened locally without the enforcement of a consistent ACL, then a user doesn't need Manager access level to edit the configuration document. Any changes they make would most likely be replicated around the network to other replicas of the database, which would not be a good situation at all for controlling usage tracking of the database.

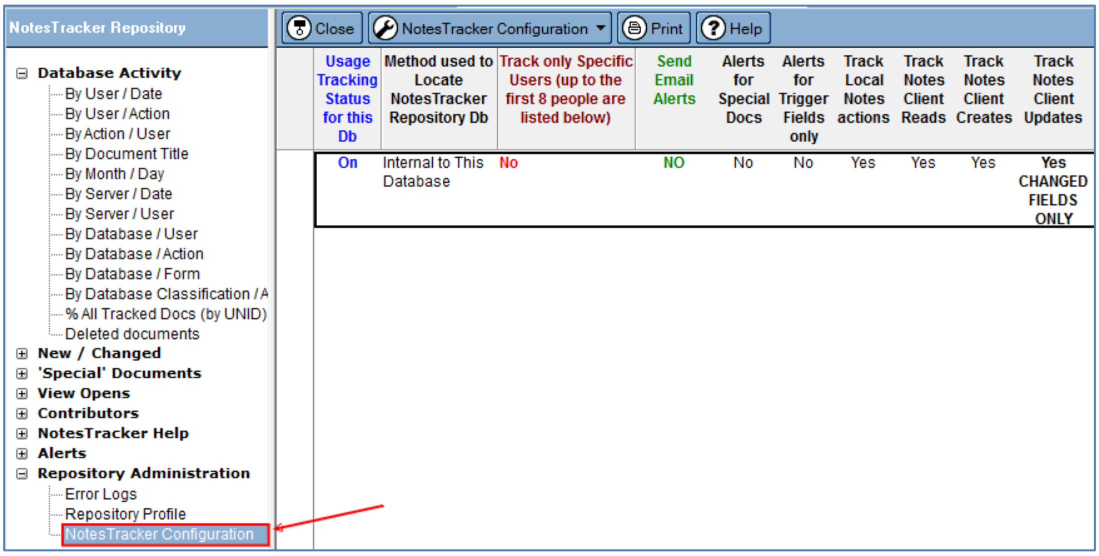
If you don't have the required Manager rights, you will see the following warning when you attempt to edit the NotesTracker Configuration Document:


You are not a Manager of this database

Sorry, you must be a MANAGER of this database in order to create and save changes to a NotesTracker configuration document.

NOTESTRACKER CONFIGURATION VIEW

The NotesTracker Configuration view can be found under the 'Repository Administration' heading on the left-hand navigation menu of the Repository database (as shown in the following image):



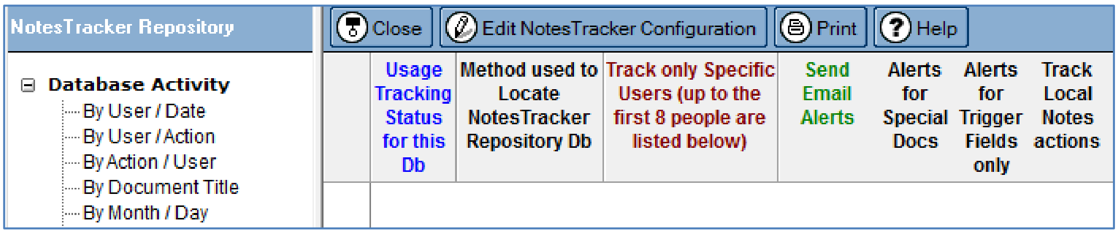
Usage Tracking Status for this Db	Method used to Locate NotesTracker Repository Db	Track only Specific Users (up to the first 8 people are listed below)	Send Email Alerts	Alerts for Special Docs	Alerts for Trigger Fields only	Track Local Notes actions	Track Notes Client Reads	Track Notes Client Creates	Track Notes Client Updates
On	Internal to This Database	No	NO	No	No	Yes	Yes	Yes	Yes CHANGED FIELDS ONLY

TIP: you should ensure that usage tracking is kept switched off for each and every NotesTracker Repository database. There is no point whatsoever in tracking usage in a NotesTracker repository. In fact, it will produce unwanted system overheads when you do such things as deleting multiple Usage Log entries (say, during an archiving operation) with tracking of document deletions active for the repository.

How you get to the NotesTracker Configuration view in each of your other Notes database applications is entirely dependent on how your Notes developer has set up the view in that database (discussed in the NotesTracker Developer Guide).

CREATING THE NOTESTRACKER CONFIGURATION DOCUMENT

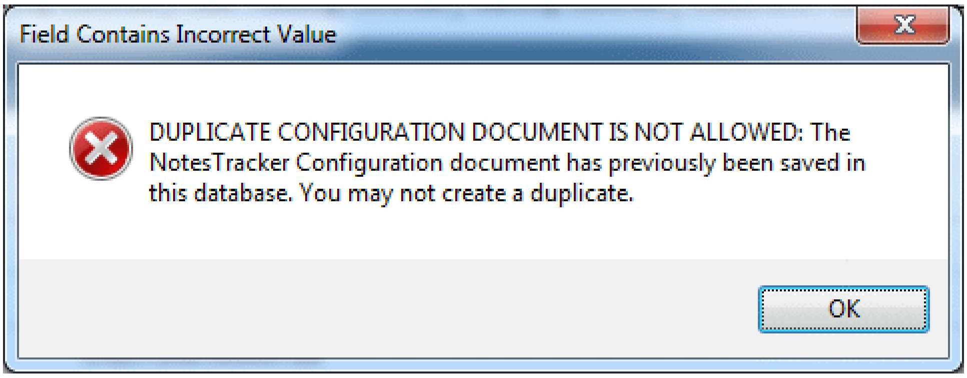
To create a new NotesTracker Configuration Document click the ‘Edit NotesTracker Configuration’ button at the top of the view.



Usage Tracking Status for this Db	Method used to Locate NotesTracker Repository Db	Track only Specific Users (up to the first 8 people are listed below)	Send Email Alerts	Alerts for Special Docs	Alerts for Trigger Fields only	Track Local Notes actions

If a NotesTracker Configuration document doesn’t already exist a new one will be created otherwise the existing configuration document will be opened for editing.

Under certain circumstances you might get the following error message immediately upon attempting to save a configuration document:

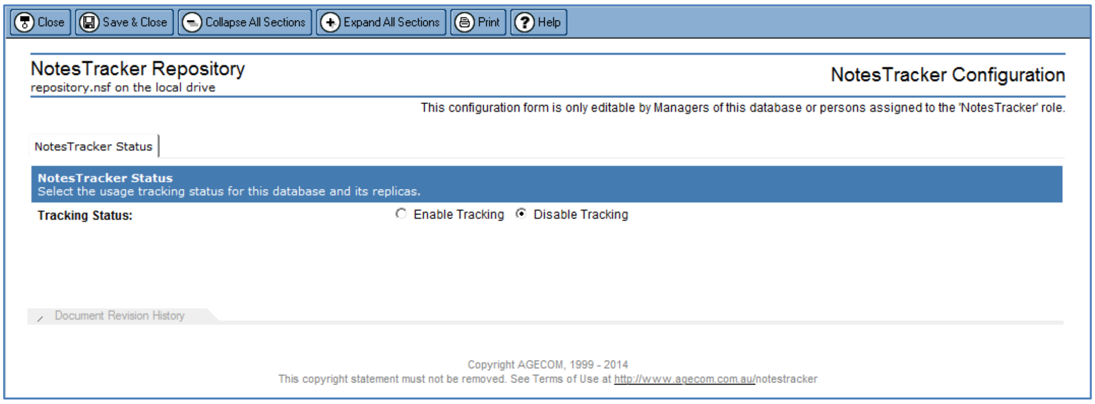


DUPLICATE CONFIGURATION DOCUMENT IS NOT ALLOWED: The NotesTracker Configuration document has previously been saved in this database. You may not create a duplicate.

If this arises, merely delete the extra configuration document(s), and then check the settings in the remaining one for completeness and accuracy. The NotesTracker code will only retrieve the “first document” in the configuration view, and if there are duplicates this might not be the one you expect.

INITIAL STATE OF THE NOTESTRACKER CONFIGURATION DOCUMENT

When you initially create the configuration document, it should look like this:



NotesTracker Repository
repository.nsf on the local drive

NotesTracker Configuration

This configuration form is only editable by Managers of this database or persons assigned to the 'NotesTracker' role.

NotesTracker Status

NotesTracker Status
Select the usage tracking status for this database and its replicas.

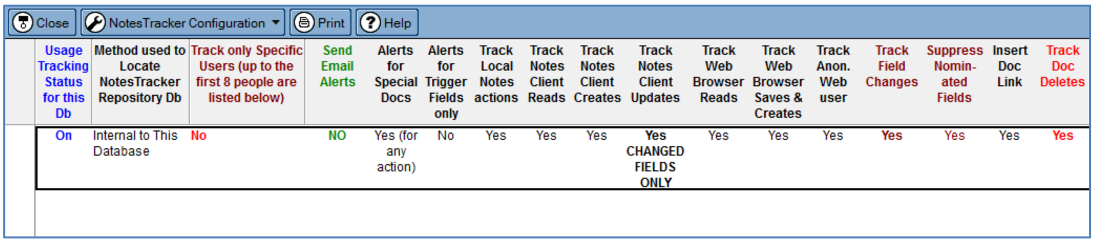
Tracking Status: ○ Enable Tracking ● Disable Tracking

Document Revision History

Copyright AGECOM, 1999 - 2014
This copyright statement must not be removed. See Terms of Use at http://www.agecom.com.au/notestracker

THE NOTESTRACKER CONFIGURATION VIEW PROVIDES A SNAPSHOT OF THE CONFIGURATION SETTINGS

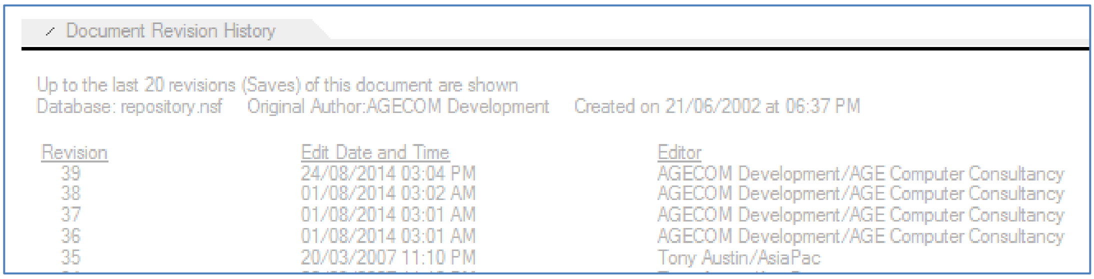
When you change the tracking status from Off to On there are quite a few options to be set. The NotesTracker configuration view shows at a glance a subset of these options:



Usage Tracking Status for this Db	Method used to Locate NotesTracker Repository Db	Track only Specific Users (up to the first 8 people are listed below)	Send Email Alerts	Alerts for Special Docs	Alerts for Trigger Fields only	Track Local Notes actions	Track Notes Client Reads	Track Notes Client Creates	Track Notes Client Updates	Track Web Browser Reads	Track Web Browser Saves & Creates	Track Anon. Web user	Track Field Changes	Suppress Nominated Fields	Insert Doc Link	Track Doc Deletes
On	Internal to This Database	No	NO	Yes (for any action)	No	Yes	Yes	Yes	Yes CHANGED FIELDS ONLY	Yes	Yes	Yes	Yes	Yes	Yes	Yes

SIMPLE EDIT HISTORY OF THE NOTESTRACKER CONFIGURATION DOCUMENT

After you have been making edits to the configuration document, you can see a simple summary of up to the last 20 configuration edits by clicking on the “twisty” at the bottom of the form, thus:



Document Revision History

Up to the last 20 revisions (Saves) of this document are shown
Database: repository.nsf Original Author:AGECOM Development Created on 21/06/2002 at 06:37 PM

Revision	Edit Date and Time	Editor
39	24/08/2014 03:04 PM	AGECOM Development/AGE Computer Consultancy
38	01/08/2014 03:02 AM	AGECOM Development/AGE Computer Consultancy
37	01/08/2014 03:01 AM	AGECOM Development/AGE Computer Consultancy
36	01/08/2014 03:01 AM	AGECOM Development/AGE Computer Consultancy
35	20/03/2007 11:10 PM	Tony Austin/AsiaPac

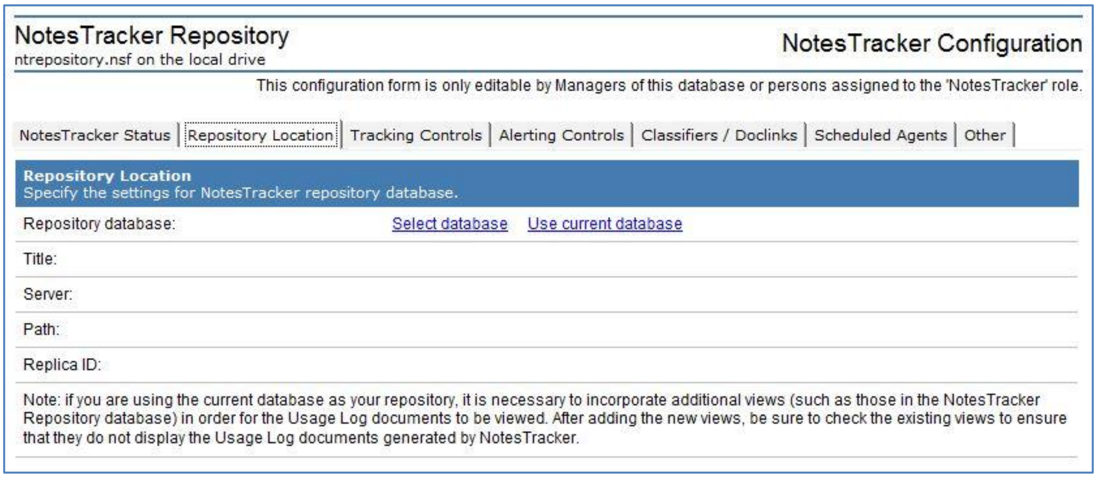
Note: if you want to, it is a simple matter to modify the design of the edit history footer by modifying the subform called **NotesTracker Simple Edit Tracking Footer**. As an example, a change that you might consider is to alter the value stored in the field “EditHistory_ListLength from 20 to some other value, which represents the number of entries in the edit history list.

If this simple edit history does not meet your requirements, it’s a simple matter to use NotesTracker to provide precise tracking of changes to its own Configuration document! (It should only take your Notes developer a few minutes to enhance the design of the configuration form.)

SPECIFYING THE NOTESTRACKER REPOSITORY LOCATION

You start the configuration process by setting the ‘Tracking Status’ option under the ‘NotesTracker Status’ tab to ‘Enable Tracking’. When tracking has been enabled additional tabs and options will automatically become available in the Configuration Document.

Click the ‘Repository Location’ tab.



NotesTracker Repository
ntrepository.nsf on the local drive

NotesTracker Configuration

This configuration form is only editable by Managers of this database or persons assigned to the 'NotesTracker' role.

NotesTracker Status | Repository Location | Tracking Controls | Alerting Controls | Classifiers / Doclinks | Scheduled Agents | Other

Repository Location
Specify the settings for NotesTracker repository database.

Repository database: Select database Use current database

Title:

Server:

Path:

Replica ID:

Note: if you are using the current database as your repository, it is necessary to incorporate additional views (such as those in the NotesTracker Repository database) in order for the Usage Log documents to be viewed. After adding the new views, be sure to check the existing views to ensure that they do not display the Usage Log documents generated by NotesTracker.

Click the ‘Select database’ option to select the appropriate Repository database or click the ‘Use current database’ to use the current database as the Repository database.

REPLICATION CONSIDERATIONS FOR LOCAL NOTESTRACKER REPOSITORIES

NotesTracker will attempt to locate the Repository database at the specified server / path. If the Repository database cannot be found NotesTracker will attempt to locate it by its Replica ID. The database may then be found on the server or on the local Notes client (if a replica has been made locally).

If the repository database cannot be found NotesTracker will put out a message on the Notes status line and then exit quietly. If the repository database can be located and opened on the user's local system, and assuming the user has the necessary Author or Depositor rights to it then NotesTracker will start logging to that local repository.

The same would happen for all users of a given application database, meaning all copies of the database having the same Replica ID and therefore the same NotesTracker Configuration document. This means that the Usage Log documents generated by NotesTracker will be distributed across the multiple workstation systems (of the multiple users of this application).

If you want centralized storage and analysis of Usage Log documents, you will need to control the proper set-up and maintenance of all users' replication settings for their local repository database in order for the dispersed Usage Log documents to be replicated to a central NotesTracker Repository.

This is just a reminder that for "local" repositories you will need to consider the administration of their replication settings. This includes such matters as scheduling of replication, sending Usage Log documents to the Domino server, whether or not you want to receive Usage Log documents from the server, preventing the users from changing the replication settings of their local repositories, and so on. Since all of this is "business as usual" for Notes/Domino administration, it is not the intention of this guide to go into detail about the matter.

CONSIDERATIONS FOR INTERNAL (SELF-CONTAINED) USAGE TRACKING

In some cases, it may be advantageous to write usage log documents "internally" in a database. For one thing, it may avoid the need to deploy a separate NotesTracker Repository database, if you haven’t already done so.

Such "self-contained usage tracking" may be a reasonable solution when one of your databases is not related in any way to other databases, and it would not make sense to write log documents for this database to the same NotesTracker Database as for others of your databases.

However, you must **allow for the increase in size of the database** caused by the writing of NotesTracker log documents into it, especially if the database is a popular one that has many operations performed on it (many document accesses – Reads, Creates, Updates, etc).

Also at least one new view must to be added to the database design or else you'll never be able to examine the log documents, and all of the existing views must be checked and modified if necessary to ignore the NotesTracker log documents.

Since this option causes the Usage Log entries to be written to the database itself, your developer has to add a view (or multiple views) to the database’s design in order for the Usage Log entries to be seen. It is pointless to select this option without any such Usage Log views. And then all of the existing views must be checked and modified if necessary to omit the NotesTracker usage log documents.

Please refer to the Notestracker Developer Guide for information on adding the required views for viewing Usage Log documents.

Additionally, users who do not have authority to create documents in the database will not cause Usage Log entries to be added internally. Nevertheless, if these design issues are resolved, it can be extremely handy to have a completely self-contained application, where the NotesTracker log entries are carried in the same database as the application documents.

For example, SDMS and CAPTURE are two popular free applications downloadable from the AGECOM website:

SDS: https://www.agecom.com.au/sds

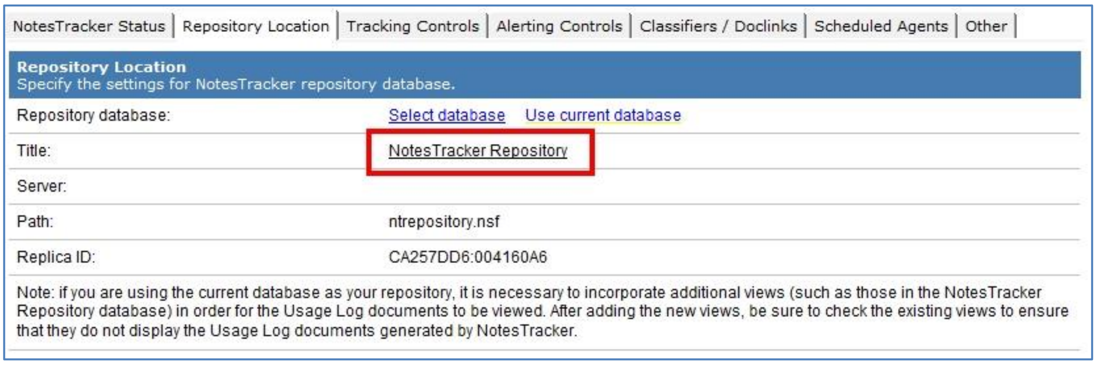
CAPTURE: https://www.agecom.com.au/capture

These applications are perfect examples of how internal usage logging can be of considerable value.

However, the designs of SDMS and CAPTURE are hidden, so it is recommended that your developers should try out the sample Customer database distributed in the NotesTracker package. This has internal logging implemented, and (in the licensed package) your developers can examine its design to see how it's done.

CHECKING THE NOTESTRACKER REPOSITORY

To make sure the NotesTracker Repository database exists and can be opened you can click on the repository database title in the NotesTracker Configuration document. After clicking on the title an attempt is made to open the repository database:


NotesTracker Status | Repository Location | Tracking Controls | Alerting Controls | Classifiers / Doclinks | Scheduled Agents | Other |

Repository Location
Specify the settings for NotesTracker repository database.

Repository database:	Select database Use current database
Title:	NotesTracker Repository
Server:	
Path:	ntrepository.nsf
Replica ID:	CA257DD6:004160A6

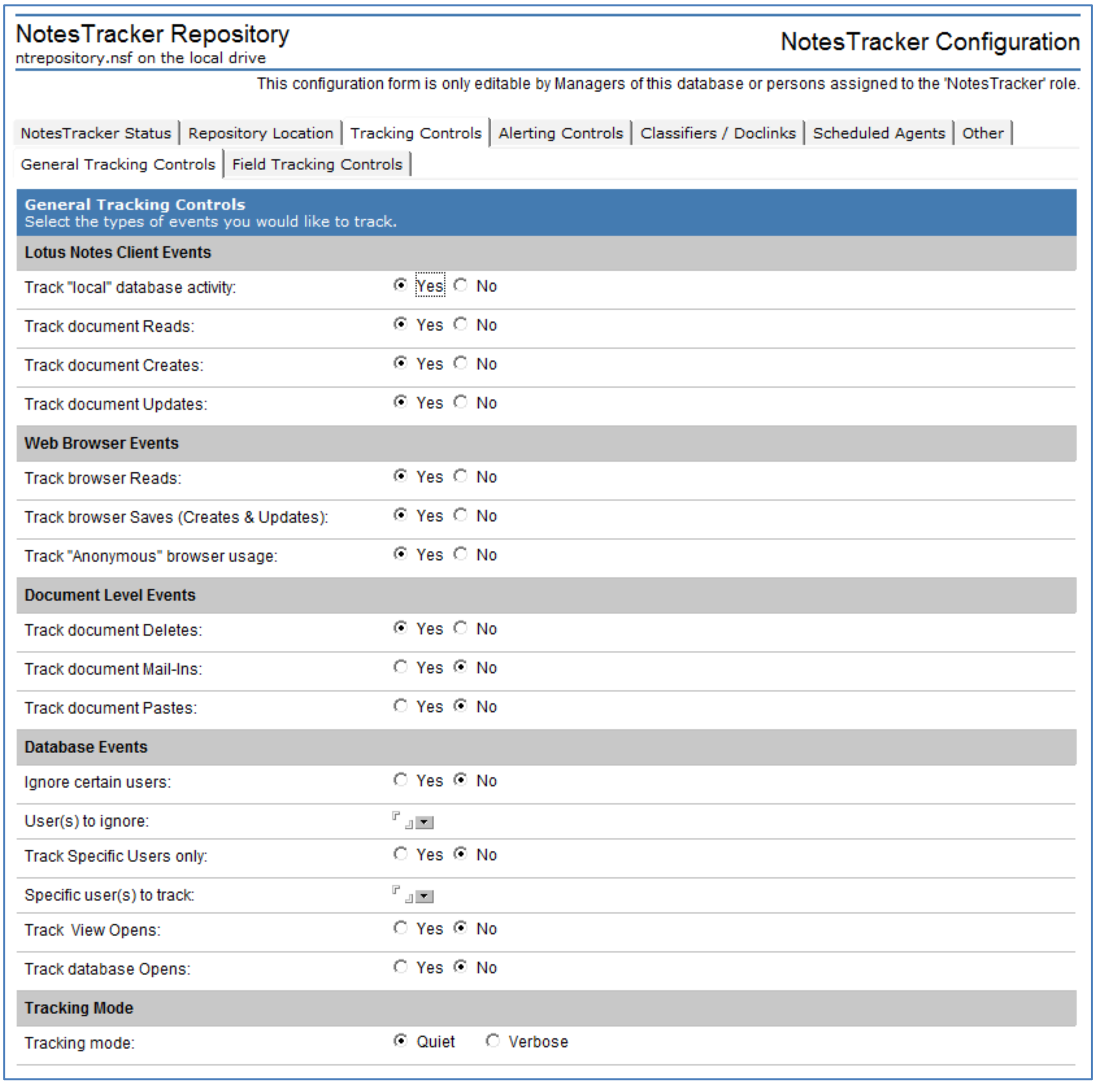
Note: if you are using the current database as your repository, it is necessary to incorporate additional views (such as those in the NotesTracker Repository database) in order for the Usage Log documents to be viewed. After adding the new views, be sure to check the existing views to ensure that they do not display the Usage Log documents generated by NotesTracker.

If the repository database is successfully found it will open in the Lotus Notes client otherwise if it cannot be found or you don't have access to it an error message will be displayed.

GENERAL TRACKING CONTROLS

Once you have defined where the NotesTracker usage log entries will be stored (external repository, or internal to the current database), you can switch to the other tabs in the configuration document and specify precisely how usage tracking should operate for the current database.

The first thing to do is to check the default settings of the 'General Tracking Controls' fields and modify them as required:



NotesTracker Repository
ntrepository.nsf on the local drive

NotesTracker Configuration

This configuration form is only editable by Managers of this database or persons assigned to the 'NotesTracker' role.

NotesTracker Status | Repository Location | Tracking Controls | Alerting Controls | Classifiers / Doclinks | Scheduled Agents | Other |

General Tracking Controls | Field Tracking Controls |

General Tracking Controls
Select the types of events you would like to track.

Lotus Notes Client Events	
Track "local" database activity:	● Yes ○ No
Track document Reads:	● Yes ○ No
Track document Creates:	● Yes ○ No
Track document Updates:	● Yes ○ No
Web Browser Events	
Track browser Reads:	● Yes ○ No
Track browser Saves (Creates & Updates):	● Yes ○ No
Track "Anonymous" browser usage:	● Yes ○ No
Document Level Events	
Track document Deletes:	● Yes ○ No
Track document Mail-Ins:	○ Yes ● No
Track document Pastes:	○ Yes ● No
Database Events	
Ignore certain users:	○ Yes ● No
User(s) to ignore:	
Track Specific Users only:	○ Yes ● No
Specific user(s) to track:	
Track View Opens:	○ Yes ● No
Track database Opens:	○ Yes ● No
Tracking Mode	
Tracking mode:	● Quiet ○ Verbose

LOTUS NOTES CLIENT EVENTS

When tracking is turned on, you can **separately control tracking** for individual document "action" types – **Read, Create, Update,** and **Delete**, and for other database events such as the opening of views document pastes.

You can **suppress logging of "local" database activity**. That is, you can switch off the tracking of databases that are opened on a desktop system's or notebook PC's local hard disk, meaning that tracking is carried out only if the database is opened from a Domino server. This may or may not be a good thing to do. It's a policy decision that is entirely up to you, the administrator. However it certainly should be approved by the application's owners/sponsors, who would go by titles such as "Knowledge Manager" or "Business Unit Executive" or "Project Manager" or "Deployment Leader" or something like that. If usage log entries are generated locally, then presumably (but not necessarily) you will arrange the database's replication local settings so that local Usage Log entries are sent to a server-based NotesTracker repository so that overall usage of the database – workstation-based plus server-based – can be consolidated and analyzed.

WEB BROWSER EVENTS

You also can **specify tracking of web browser actions: browser Reads, and browser Saves (Creates and Updates).** And as a means of reducing usage tracking overheads in a busy (high transaction rate) Internet situation, you can **suppress the logging of "Anonymous" web user activities** meaning that only the actions performed by authenticated browser users will be tracked (those users who can provide an acceptable user name and password when challenged and who most likely will be far fewer in number than anonymous users).

DOCUMENT-LEVEL EVENTS

You can also track document-level actions, meaning the sort of actions that occur on a document as a whole without its having to be opened. Also, such documents might take place against a set of documents and not just against a single document at a time. These action types are **Deletes, Pastes and Mail-ins**.

DATABASE EVENTS

Finally, there are database-level actions: **View Opens** and **Database Opens**. These are unrelated to any individual documents at all.

The tracking of *view opens* is meant to be a very short-term activity that will gives your Notes developer or administrator some idea of the relative usage of views in a given database, so that there is a rational basis for changing the properties and/or designs of the views in order to give better view performance and/or reduce Domino server overheads (processor and view index space).

The tracking of *database opens* is of similar interest. It can give you an overall feel for the popularity of a database. For what it's worth, note that if you are also tracking view opens naturally enough there is some tracking overlap between the two, because the database's default view is opened then too.

The original intent of NotesTracker was to track the database accesses for *all users,* but it was decided to enable a subset of users to be tracked, in whatever circumstances that you may not want or need all users to be tracked. It is recommended that this restriction to a subset of users be used judiciously, especially where it would render ineffective your attempts to monitor and audit all database activity for legal, criminal, or privacy-related reasons.

IGNORING USAGE TRACKING FOR CERTAIN USERS

This feature provides greater flexibility in control of which users of a database are tracked. Any users whose names are listed in this field will **not** be tracked.

If you select “**Yes**“ for the “**Ignore certain users**“ option then you are required to enter at least one user’s name. Each name entered must resolve to a valid Notes name format, such as “John Smith/Acme” or “Jacques La Salle/Sales/France” (the so-called “abbreviated” name format). Be very careful to enter names in this format. Tracking will not occur if a user’s Notes name is misspelled. To assist with this, an address dialog is provided so that you can select the name(s) from your address book(s).

Note: this option takes precedence over the option, described next, for restricting usage tracking to specific users. Therefore if a user's name appears in both lists, that user's actions will not be tracked.

RESTRICTING USAGE TRACKING TO SPECIFIC USERS

If you select “**Yes“** for the “**Track Specific Users only**“ option then you are required to enter at least one user’s name. As above, each name entered must resolve to a valid Notes name format.

A variation of this rule is that the name “**Anonymous**” may be used if, for some reason, you wish to track usage by unauthenticated Web browser users. (The name has to be *typed* into the “Track Specific Users only” field, since it is not available in the address dialog.)

Note: the test as to whether or not a user is specified to be tracked is made very early in each monitored event, to help minimize usage tracking overheads.

TRACKING MODE

Quiet and Verbose tracking mode are explained a page or two later in this guide.

NOTESTRACKER PERFORMANCE CONSIDERATIONS

Naturally, any sort of usage tracking will have some performance overheads. In the design and coding of NotesTracker, every attempt has been made to minimize these overheads. Tips and guidelines about the performance implications of the various settings can be viewed by clicking the ‘Help’ button at the top of the configuration settings form.

As a rule, the great bulk of operations against a database (80 percent or more in most cases) are document reads via Notes Clients or Web browsers. Therefore setting **Track document reads** and **Track browser reads** to “**No**” will have a great effect in reducing tracking overheads.

Similarly, setting **Track “Anonymous” browser reads** to “**No**” could significantly reduce the browser usage tracking overheads, especially if the database is getting a large number of hits across the Internet with most of the database accesses being carried out by unauthenticated browser users. This setting still causes tracking of authenticated browser users (where you have set the database’s ACL to force the user name/password challenge to appear when an attempt is made to open the database via a browser). Presumably you will want to track whatever operations the authenticated users are carrying out, so this setting is seen as providing a reasonable compromise between reducing performance overheads and obtaining useful tracking information.

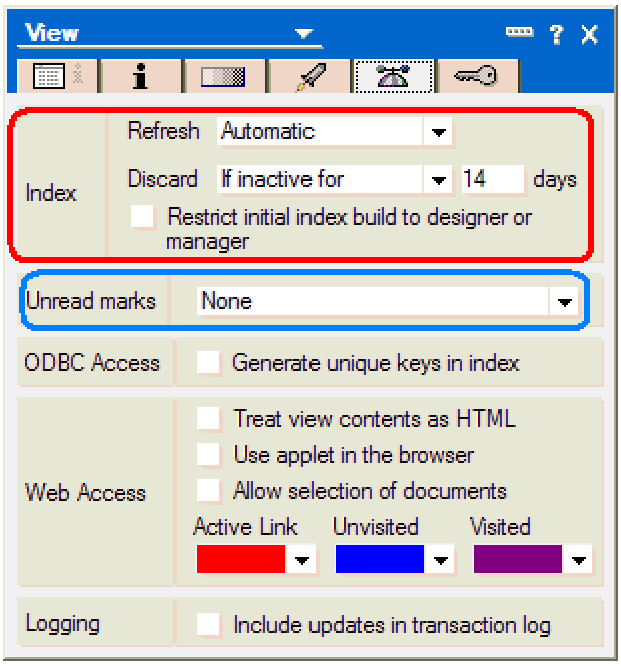
The default settings presume that you will always want to track document deletions, but not always to track view opens, database opens, document mail-ins, and document pastes.

It is fairly common for document delete and paste operations to be performed against *groups of documents*, so if you are tracking deletes and pastes keep this in mind.

The ability to track **view opens** was added to an early release of NotesTracker to provide (we believe) a unique yet simple way for Domino administrators and developers to determine which views in a database are being most heavily used and which ones are rarely used. It should only take a few hours, or a few days at most, to determine this, because users tend to switch views very frequently. As soon as you have gathered sufficient view usage statistics, you definitely should switch off view tracking.

Given an insight into view usage you might decide to entirely eliminate some views or at least to consolidate some of them.

As well, and you should review and if necessary change their properties which impact performance, such as index refresh and discard intervals, and whether to keep unread marks:



Index
Refresh Automatic
Discard If inactive for 14 days
□ Restrict initial index build to designer or manager

Unread marks None

ODBC Access □ Generate unique keys in index

Web Access
□ Treat view contents as HTML
□ Use applet in the browser
□ Allow selection of documents
Active Link 🟥 Unvisited 🟦 Visited 🟪

Logging □ Include updates in transaction log

Of course, switching off some of the above tracking options would not be feasible if it eliminates exactly the type of usage tracking that you want to carry out against the database. In this case, you might consider adopting a “sampling approach” such as carrying out view tracking once off (or only occasionally) or for short periods of time, say one day per month.

VERBOSE AND QUIET TRACKING MODES, PLUS SOME TYPICAL TRACKING ISSUES

As a licensed user with access to the source code, if you examine the various LotusScript routines you will find that wherever possible error situations have been foreseen and handled in the code. Our own research, together with feedback and suggestions from NotesTracker users worldwide, have gotten NotesTracker to the stage where most error conditions are handled gracefully – that is, no surprises!

The key design philosophy behind NotesTracker's handling of errors is that when an error occurs anywhere in the tracking routines the end user should not notice any difference whatsoever in the behavior of the application.

For example, if the NotesTracker configuration view or the configuration document is missing or if the specified external NotesTracker Repository database cannot be opened, the code detects this and handles it "quietly" in order to cause minimal or no disruption to the normal operation of an application.

TRACKING MODE

This section applies only for database accesses via a Notes Client.

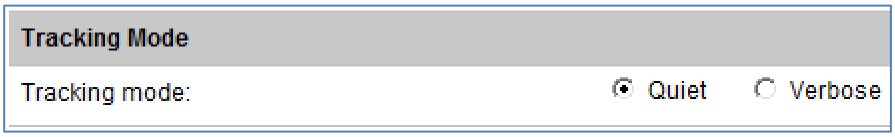
NotesTracker sends occasional status and all of its error messages only to the **Notes Client status line** (at the bottom of the Notes Client window), rather than displaying disruptive dialog boxes.

If you find that usage tracking is not occurring at all for a given database, or that some but not all of the selected usage tracking options seem to be working, you should examine the messages that appear on the status line.

A small number of additional messages pop up when you click on the status line. Since the status line has a limited buffer (only about ten or fifteen messages) you'll need to be quick to catch any relevant messages.

To be non-disruptive to the normal operation of your Notes applications that are being tracked, NotesTracker does not display any dialog boxes. This is the "Quiet" mode, when only an occasional advisory message appears on the status line.

There is a "Tracking Mode" setting in the NotesTracker Configuration Document. Its purpose is to provide an intermediate level of troubleshooting and problem solving or each database:



Tracking Mode

Tracking mode: ● Quiet ○ Verbose

If you switch to Verbose mode, then a very modest range of additional advisory and error messages can appear in the Notes Client status line. Careful monitoring of these additional messages should help you to diagnose and rectify some NotesTracker problems. The messages are generated via Print statements (in the LotusScript code), which cause the messages to be sent to the status line.

Be sure to switch the setting back to "Quiet" mode when you've finished your troubleshooting.

If the status line messages do not help you to clear up the tracking problem, then it's time to call in your Notes developer (the one who's supporting NotesTracker in your organization) to carry out some in-depth problem analysis for usage tracking on that particular database. If you see fit, you might drop in some additional Print statements at critical points in the NotesTracker code to provide you with a more “aggressive” style of error reporting. (A reminder: keep a record of your additional statements, and where you have placed them, so that you can re-enter them in the same spots following future releases of NotesTracker.)

Of course, you can always contact us at AGECOM for additional support.

STATUS LINE MESSAGES RELATED TO DEVELOPMENT ISSUES

The messages might indicate that the developer has not set up NotesTracker correctly for the database, such as forgetting to add a required design element (like a form or view). Problems like this should have been eliminated by careful testing prior to deployment, and really shouldn’t occur in the production environment!

STATUS LINE MESSAGES RELATED TO OPERATIONAL ISSUES

Problems that might occur in the production environment are generally due to operational issues. These are generally related to incorrect settings in the NotesTracker Configuration Document, such as:

- Incorrectly specifying the NotesTracker repository database’s Replica ID or Server and Path names

- The NotesTracker Configuration Document being inadvertently deleted from the database.

- The NotesTracker Configuration suffering from corruption (rare, but possible; and probably any corruption in the database will cause other operational problems that are far more troubling than the effects of corruption on usage tracking).

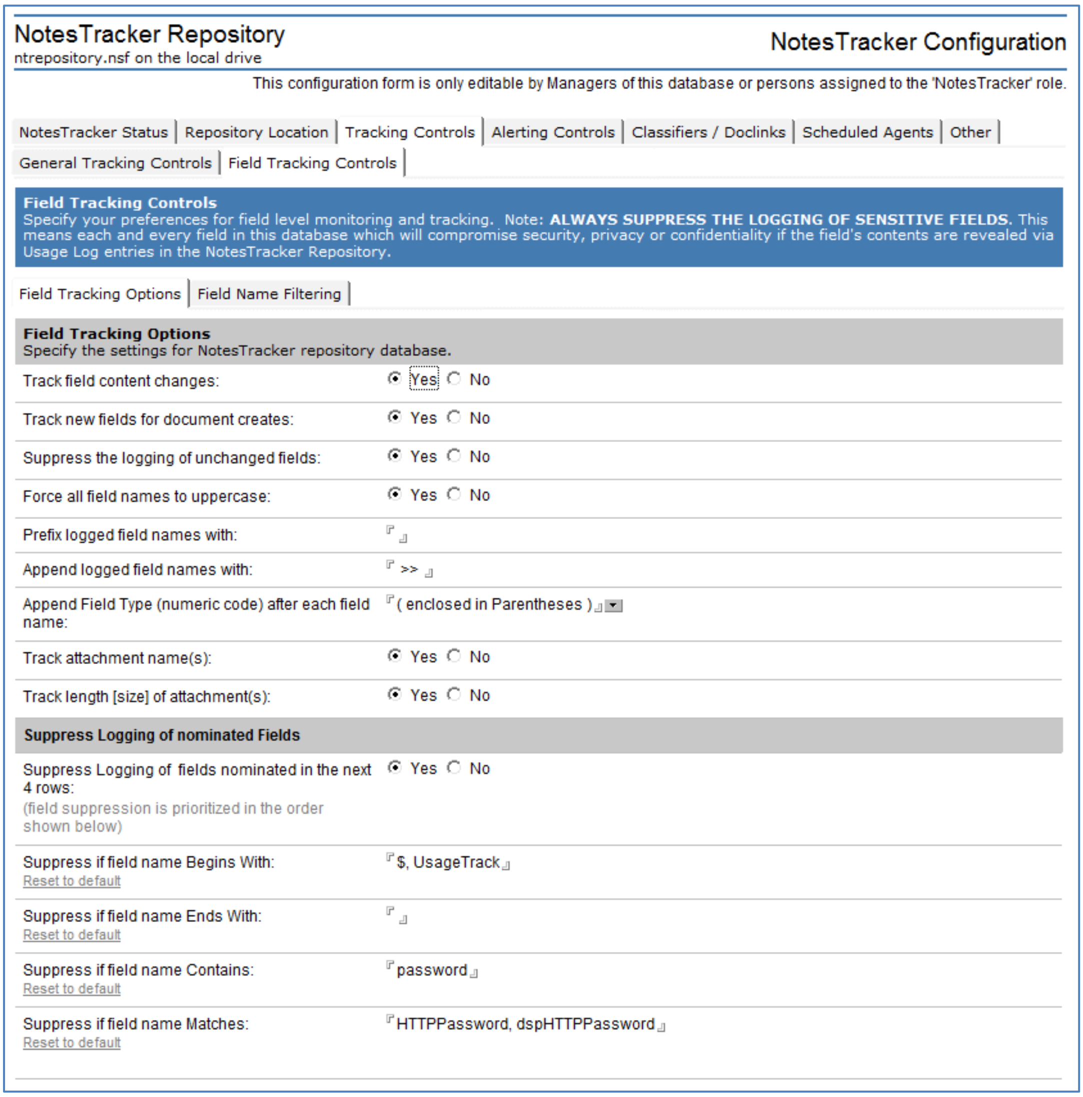
- The NotesTracker Configuration Document undergoing replication problems, leading to multiple configuration documents. Only the settings in the first document in the configuration view will be honored. (The superfluous configuration documents should be removed, with care, from the database and the settings in the remaining configuration document should be reviewed for accuracy.)

TRACKING DOCUMENT UPDATES – FIELD TRACKING CONTROLS

NotesTracker is all about tracking what happens to the content of your database applications. A key component of the content is the array of **fields** stored within documents, and this section of the configuration document enables you to specify in considerable detail how fields are to be tracked.

FIELD TRACKING OPTIONS

If the option ***Track Document Updates*** in the General Tracking Controls tab is set to "**Yes**" then (as shown in the following figure) under the **Field Tracking Controls** tab you can switch on the tracking of fields by setting the option **'Track FIELD CONTENT CHANGES'** to "**Yes**" (the default value):



NotesTracker Repository
ntrepository.nsf on the local drive

NotesTracker Configuration

This configuration form is only editable by Managers of this database or persons assigned to the 'NotesTracker' role.

NotesTracker Status | Repository Location | Tracking Controls | Alerting Controls | Classifiers / Doclinks | Scheduled Agents | Other
General Tracking Controls | Field Tracking Controls

Field Tracking Controls
Specify your preferences for field level monitoring and tracking. Note: **ALWAYS SUPPRESS THE LOGGING OF SENSITIVE FIELDS.** This means each and every field in this database which will compromise security, privacy or confidentiality if the field's contents are revealed via Usage Log entries in the NotesTracker Repository.

Field Tracking Options
Specify the settings for NotesTracker repository database.

Track field content changes:	● Yes ○ No
Track new fields for document creates:	● Yes ○ No
Suppress the logging of unchanged fields:	● Yes ○ No
Force all field names to uppercase:	● Yes ○ No
Prefix logged field names with:	
Append logged field names with:	>>
Append Field Type (numeric code) after each field name:	(enclosed in Parentheses)
Track attachment name(s):	● Yes ○ No
Track length [size] of attachment(s):	● Yes ○ No
Suppress Logging of nominated Fields	
Suppress Logging of fields nominated in the next 4 rows: (field suppression is prioritized in the order shown below)	● Yes ○ No
Suppress if field name Begins With: Reset to default	$, UsageTrack
Suppress if field name Ends With: Reset to default	
Suppress if field name Contains: Reset to default	password
Suppress if field name Matches: Reset to default	HTTPPassword, dspHTTPPassword

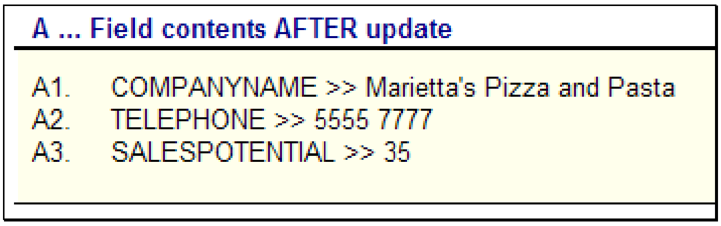
The default value of “**Yes**” for the second option “**Suppress the logging of UNCHANGED FIELDS**” causes NotesTracker to suppress the logging of the contents of unchanged fields (that is, where a field's “After Image” is the same as its “Before image”). If you alter this option to “**No**” then NotesTracker stores the values of all of the fields in a document, not just of the changed fields (defined as the new, updated and deleted fields).

Many Notes databases after some time at the hands of a range of developers finish up with a higgledy-piggledy collection of fields some with field names in upper case, some in lower case, and some in mixed case! In order for a Usage Log document to present the field names uniformly, there is a default value of “**Yes**” for the option “**Force all field names to UPPER CASE**”. This forces each field name to be converted to upper case in the Usage Log documents, which (as well as uniformity of presentation) can help each field name to stand out from the field’s contents.

IMPROVING THE PRESENTATION OF USAGE LOG “BEFORE” AND “AFTER” FIELD VALUES

Also for better presentation purposes, the next option “**Prefix logged field names with**” allows you to provide a string of "eye catcher" characters as a prefix to each field name. This prefix could be put to any use – say, putting a Run Number, project identifier, or other meaningful text string in front of each field name. The default prefix is blank.

Via the option “**Append logged field names with**” you should specify a string to be placed (appended) after each field name in the Usage Log document. The intention is to cause field names in the Usage Log to stand out from the field contents:

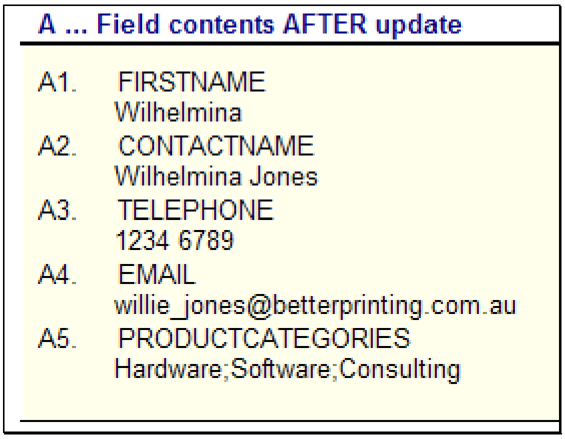


A ... Field contents AFTER update

A1. COMPANYNAME >> Marietta's Pizza and Pasta
A2. TELEPHONE >> 5555 7777
A3. SALESPOTENTIAL >> 35

The default suffix is a chevron pattern: “ **>>** “ (notice the leading and trailing single space characters). The suffix could be a null character (an empty value), but should be at least a single blank character.

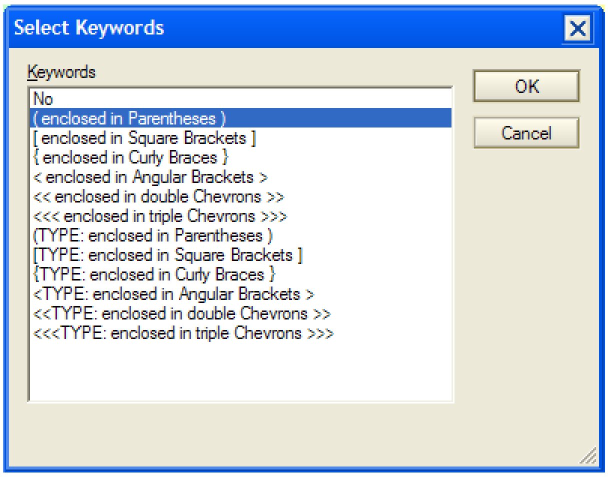
A special option for the suffix is to specify ***NL** which stands for “New Line” (do not omit the leading asterisk). This causes the field name to display on one line and the field contents to display starting at the beginning of the next line. For example:



A ... Field contents AFTER update

A1. FIRSTNAME
Wilhelmina
A2. CONTACTNAME
Wilhelmina Jones
A3. TELEPHONE
1234 6789
A4. EMAIL
willie_jones@betterprinting.com.au
A5. PRODUCTCATEGORIES
Hardware;Software;Consulting

Another option is “**Append FIELD TYPE (numeric code) after each field name**” which places after each field name an integer value that represents the data type of the field. You also have the ability to specify via a pull-down list how the data type integer is displayed, either plain (no enclosing characters) of with a range of different enclosing characters (the default being enclosing parentheses), thus:



Select Keywords

Keywords

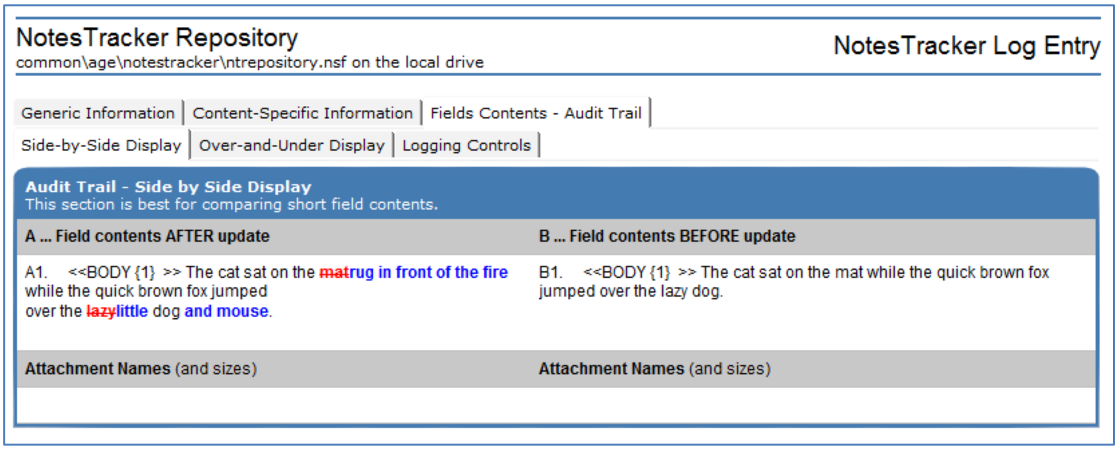
No
(enclosed in Parentheses)
[enclosed in Square Brackets]
{ enclosed in Curly Braces }
< enclosed in Angular Brackets >
<< enclosed in double Chevrons >>
<<< enclosed in triple Chevrons >>>
(TYPE: enclosed in Parentheses)
[TYPE: enclosed in Square Brackets]
{TYPE: enclosed in Curly Braces }
<TYPE: enclosed in Angular Brackets >
<<TYPE: enclosed in double Chevrons >>
<<<TYPE: enclosed in triple Chevrons >>>

"BEFORE VALUES" AND "AFTER VALUES" OF FIELDS

The following illustrations (taken from a NotesTracker Log Entry document) show the effect of default field name suffix being used, that is, the chevron string " >> " (a space followed by two Greater Than symbols followed by another space), and with the field type option is switched on.

To see the field changes click on the **Field Contents - Audit Trail** tab.

Side-by-Side Display



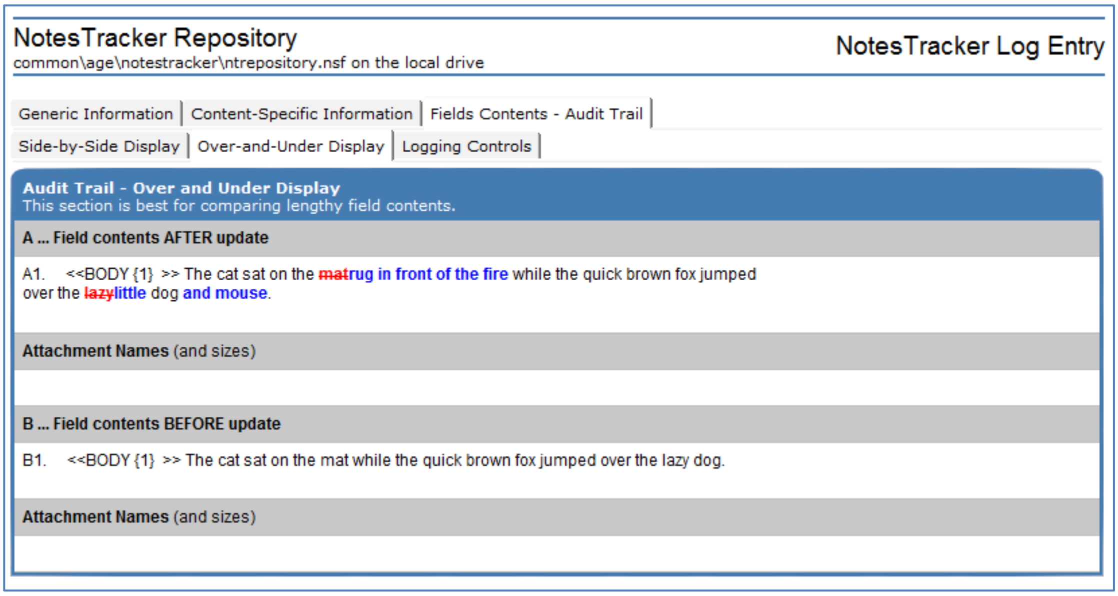
NotesTracker Repository
common\age\notestracker\ntrepository.nsf on the local drive

NotesTracker Log Entry

Audit Trail - Side by Side Display
This section is best for comparing short field contents.

A ... Field contents AFTER update	**B ... Field contents BEFORE update**
A1. <<BODY {1} >> The cat sat on the **~~mat~~rug in front of the fire** while the quick brown fox jumped over the **~~lazy~~little** dog **and mouse**.	B1. <<BODY {1} >> The cat sat on the mat while the quick brown fox jumped over the lazy dog.
Attachment Names (and sizes)	**Attachment Names** (and sizes)

Over-and-Under Display



NotesTracker Repository
common\age\notestracker\ntrepository.nsf on the local drive

NotesTracker Log Entry

Audit Trail - Over and Under Display
This section is best for comparing lengthy field contents.

A ... Field contents AFTER update

A1. <<BODY {1} >> The cat sat on the **~~mat~~rug in front of the fire** while the quick brown fox jumped
over the **~~lazy~~little** dog **and mouse**.

Attachment Names (and sizes)

B ... Field contents BEFORE update

B1. <<BODY {1} >> The cat sat on the mat while the quick brown fox jumped over the lazy dog.

Attachment Names (and sizes)

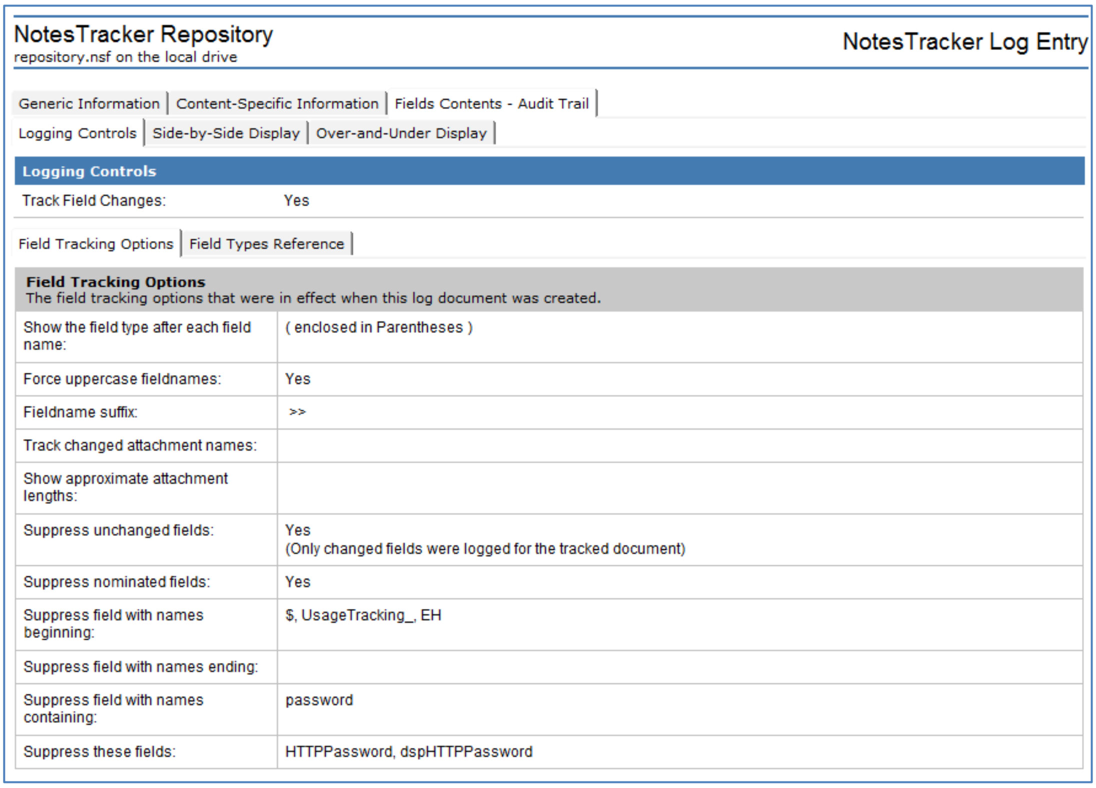
The "Side by Side" and "Over and Under" sections provide alternatives for the display of the field contents. Select the one that is best for displaying the particular contents of a given document.

You can easily see the changes made. Content that has been added is highlighted in bolded blue text. Content that has been removed is highlighted in bolded red text with a strikethrough mark through it.

Note: for your convenience in analyzing the usage log entries, the field names are prefixed with a sequentially incremented field identifier (for “Before Images” B1, B2, B3, … and for “After images” A1, A2, A3, … where of course “B” stands for “Before” and “A” for “After”). You should understand that due to the way that the field’s data is gathered by NotesTracker, by looping through all of the field items in a document, there isn’t a guaranteed one-to-one relationship between these sequence numbers. Thus the field labeled A1 is not necessarily the same as the one labeled B1, and so on. Nevertheless, the sequence numbers often do finish up being identical, or at worst fairly close together.

If you specify the special value of ***NL** (an asterisk followed by an uppercase N and an uppercase L) in the NotesTracker Configuration document, then a New Line character is inserted between the field name and the field contents. Using this value in conjunction with "Force all field names to UPPER CASE" can lead to better legibility in the Usage Log document.

The ‘*Field Contents – Audit Trail → Logging Controls → Field Tracking Options*’ section of the NotesTracker Log Entry document lists the field tracking options that were in effect in the database at the instant the document was opened. (Remember that NotesTracker reads the configuration document for the database for each and every logged event).



NotesTracker Repository
repository.nsf on the local drive

NotesTracker Log Entry

Generic Information | Content-Specific Information | Fields Contents - Audit Trail
Logging Controls | Side-by-Side Display | Over-and-Under Display

Logging Controls

Track Field Changes: Yes

Field Tracking Options | Field Types Reference

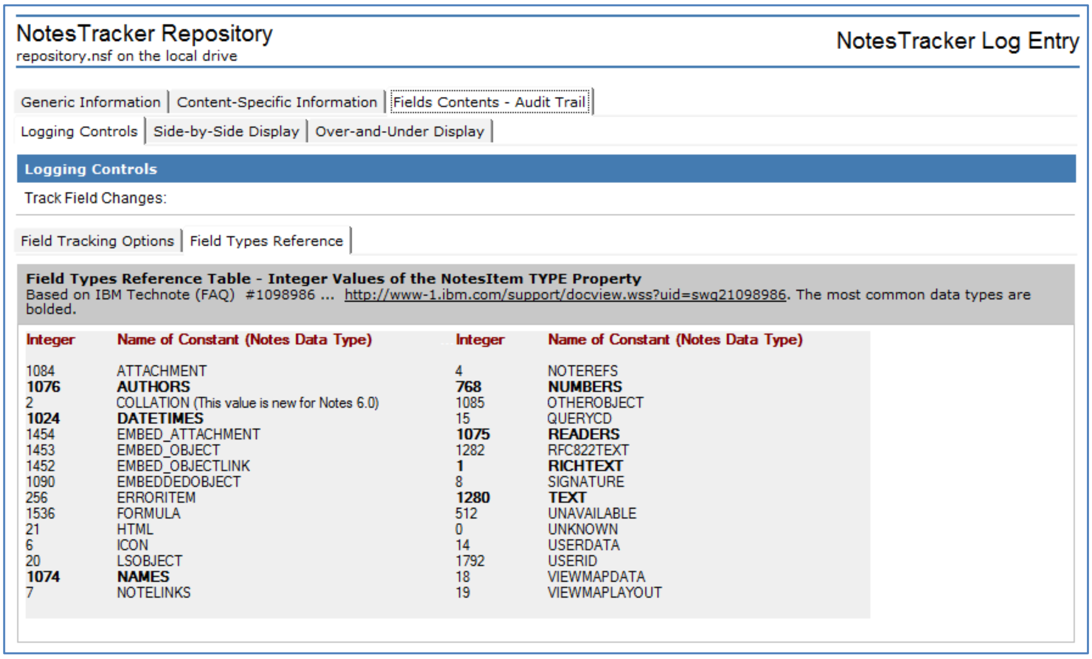
Field Tracking Options
The field tracking options that were in effect when this log document was created.

Show the field type after each field name:	(enclosed in Parentheses)
Force uppercase fieldnames:	Yes
Fieldname suffix:	>>
Track changed attachment names:	
Show approximate attachment lengths:	
Suppress unchanged fields:	Yes (Only changed fields were logged for the tracked document)
Suppress nominated fields:	Yes
Suppress field with names beginning:	$, UsageTracking_, EH
Suppress field with names ending:	
Suppress field with names containing:	password
Suppress these fields:	HTTPPassword, dspHTTPPassword

FIELD TYPES

When the option “**Append FIELD TYPE (numeric code) after each field name**” has been enabled in the NotesTracker Configuration document using the default value of “**(enclosed in parentheses)**” you will see the numeric field type appended to the field names in the NotesTracker Log Entry documents.

The field type description for each numeric value can be found by selecting the ‘*Field Contents – Audit Trail → Logging Controls → Field Types Reference*’ section of the NotesTracker Log document, as in the following image. The data types (name of constants) are arranged in ascending name order (not numerical value):



NotesTracker Repository
repository.nsf on the local drive

NotesTracker Log Entry

Logging Controls

Track Field Changes:

Field Types Reference Table - Integer Values of the NotesItem TYPE Property
Based on IBM Technote (FAQ) #1098986 ... http://www-1.ibm.com/support/docview.wss?uid=swq21098986. The most common data types are bolded.

Integer	Name of Constant (Notes Data Type)	Integer	Name of Constant (Notes Data Type)
1084	ATTACHMENT	4	NOTEREFS
1076	**AUTHORS**	**768**	**NUMBERS**
2	COLLATION (This value is new for Notes 6.0)	1085	OTHEROBJECT
1024	**DATETIMES**	15	QUERYCD
1454	EMBED_ATTACHMENT	**1075**	**READERS**
1453	EMBED_OBJECT	1282	RFC822TEXT
1452	EMBED_OBJECTLINK	**1**	**RICHTEXT**
1090	EMBEDDEDOBJECT	8	SIGNATURE
256	ERRORITEM	**1280**	**TEXT**
1536	FORMULA	512	UNAVAILABLE
21	HTML	0	UNKNOWN
6	ICON	14	USERDATA
20	LSOBJECT	1792	USERID
1074	**NAMES**	18	VIEWMAPDATA
7	NOTELINKS	19	VIEWMAPLAYOUT

Note: it was decided to use the numeric value for field types (rather than name of the constant) to conserve space.

Knowing the field type for each field can be very useful (even necessary) in some situations, and could help with data fidelity management or application debugging. The above illustration would indicate, for example, that the telephone “number” field is actually of type text (1280) rather than numeric (768).

LIMITS FOR HOW FIELD “BEFORE” AND “AFTER” VALUES ARE STORED

The “before” values are stored in the Usage Log document in a single plain Text field named **DocActivity_Field_BeforeValues** and the “after” values in a Text field named **DocActivity_Field_AfterValues**.

The design assumption made early in the life of NotesTracker was that the total number of characters used to display each of the Before and After values fields – including the additional characters used for presentation control (described in the previous few pages – does not reach the 32K characters length limit for a single Text field.

Supposing that each logged field on average takes up about 15 character positions, this means approximately 2,000 fields could be logged. Or if each field takes up about 20 positions, then roughly 1,500 fields could be logged. For most users this represents adequate headroom. At the time of writing (NotesTracker v5.3), no user of NotesTracker has ever raised this as an issue.

You could contact AGECOM if your field names field data is reaching this 32K limit and you need further guidance in this matter. One way of bypassing this limit would be to define that only nominated fields be logged, rather than all. An alternative would be to alter these fields in the Usage Log form to Rich Text type, but this workaround would take design, coding and testing effort on your part.

CONSIDERATIONS FOR TRACKING DATABASE ACCESSES VIA WEB BROWSERS

A major feature incorporated into NotesTracker is the option to track “front end” accesses against your Notes databases carried out via web browsers.

You can separately specify whether you want tracking of **web browser Reads** and **web browser Saves**.

A web browser **Read** is logged every time that a web page is posted out from your Domino server to a browser user.

A web browser **Save** is logged every time that a web page is returned from the browser to your Domino server. Web saves cannot in and of themselves distinguish between the **creation** and the **update** of a Notes document.

This is in distinct contrast with the Notes Client environment where document creates and update events can be detected separately and logged as such. The Lotus Notes Client environment is "stateful" or "state-aware". A long- running session is established between the Domino server and the Notes Client. The session lasts from sign on to sign off, and the Domino server is able to associate each read/write operation with a specific Notes Client user session. A fundamental design decision for NotesTracker was to record only a single event when a new document is created or an existing document is updated, no matter how many times the document is saved during the time that the document is open.

In contrast, the web browser environment is "stateless" -- the Domino server POSTs (sends) a web page to browsers, and GETs (receives) web pages back from browsers. There is no simple way to associate the field values in an outgoing web page with the field values that are received back some time later from the browser window.

A limited degree of field change tracking might have been possible via browser "cookies" (sometimes called "session cookies" because they are used to simulate session awareness that is missing from the HTTP protocol). However, this would have involved intricate programming for NotesTracker itself, and changes to many of your database forms) which was felt not to be justified for NotesTracker. More importantly, using cookies for such a purpose is not a reliable mechanism for tracking field changes, because an individual browser user can disallow the use of cookies (this being a personal decision, or maybe a corporate requirement).

The field “After Images” are logged every time that a user clicks the SUBMIT button in the browser window so sending the page back to the Domino server.

The field "Before images" are logged too. In order to determine what fields are changed, the before images are obtained by retrieving their values from the so-called “back end” Notes document (that is, by again retrieving the saved document from the database), only then having a set of before images to compare with the after images stored in the current web page.

It is important to note that the “back end” Notes document so retrieved just might have been updated in the intervening period – seconds, minutes or even hours – since the web page was originally posted to the user’s browser window, (In the Notes Client environment the before images are recorded as soon as the document is opened, and no matter how many times you save the document it is only when the document is finally closed that the field after images are recorded. This is possible because the session between the Notes Client and the Domino server is “stateful” or “state aware”.)

That all being said, testing shows that in many if not most cases NotesTracker's web browser update Usage Log entry seems to give an accurate representation of the actual field changes.

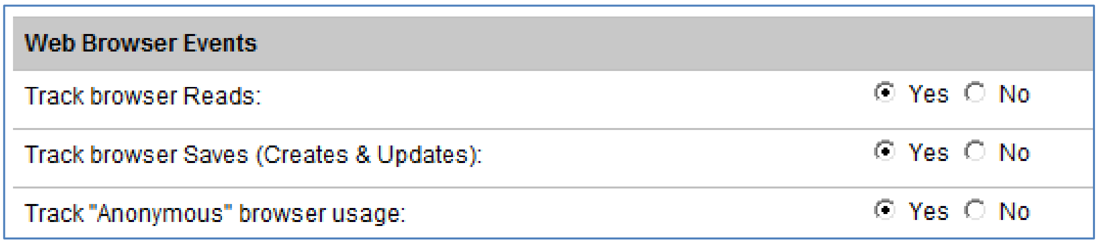
WEB BROWSER TRACKING – PERFORMANCE CONSIDERATIONS

It is wise to keep **performance considerations** in mind at all times. This is particularly so for Internet accesses to your databases, which are outside your control and might place very heavy loads on your Domino server.

The great bulk of web browser accesses to your Notes databases almost certainly will be performed by "the general public" -- non-authenticated users, to whom Domino gives the user name **"Anonymous"**.

You may be able to significantly reduce usage tracking overheads (CPU load, memory and disk space on your Domino servers) by specifying that you do not want anonymous user accesses to be tracked. Naturally enough, if you really do need to know about anonymous access, this will reduce the subsequent analytical value of the usage logs. In some cases (such as where you only allow authenticated browser access against specific Notes databases by your staff, your customers, your suppliers or your business partners), an idea may be to retain the logging of anonymous accesses but only for Web Page Saves (creates and updates) and not for Reads.

A section in the NotesTracker Configuration Document gives you this degree of control is as follows:



Web Browser Events	
Track browser Reads:	● Yes ○ No
Track browser Saves (Creates & Updates):	● Yes ○ No
Track "Anonymous" browser usage:	● Yes ○ No

Note: remember that these settings apply to individual Notes databases, giving you a fine degree of control on a database-by-database basis. This contrasts with use of the regular Domino Web Server Log database (DomLog.NSF) used by other web tracking solutions.

With NotesTracker you have the option of logging all of the field contents ("After images") associated with each new or updated Notes document, but DomLog.NSF does not record this level of detail.

It is our opinion that DomLog.NSF and NotesTracker provide complementary web tracking mechanisms. As far as we are aware, only NotesTracker provides a generic Notes Client activity tracking mechanism.

LIMITATIONS FOR THE LOGGING OF RICH TEXT FIELD CONTENTS, INCLUDING ATTACHMENTS

DIFFICULTIES IN TRACKING RICH TEXT FIELD CHANGES

The contents of Rich Text fields (such as the commonly-occurring "Body" field in mail memos) are not fully logged by NotesTracker.

Only a simplified **plain-text rendition** of a Rich Text field’s contents is written to the Usage Log document. Attachments and embedded objects are not captured by NotesTracker, nor are text styles (such as bold, underlined and italic) retained.

This was a deliberate design decision, starting with the very first NotesTracker release. It was felt to be unnecessary and beyond the purposes and scope of NotesTracker to log changes to the complex structure of Rich Text fields.

Firstly, it was regarded as undesirable to make log copies of a document's attachments, since it was likely that as this could vastly increase the disk space requirements for the NotesTracker Repository database.

Secondly, it was felt that the logging of changes to embedded objects, such as Word Processing documents or spreadsheets, would be a considerable programming challenge that was not justified at the time.

Thirdly, it was thought that it would be sufficient for most purposes to log just a simple text (plain text) rendition that shows how the contents of a Rich Text field have changed. Or, putting this another way, the assumption was made that most users of NotesTracker would be satisfied to record just the textual changes made to Rich Text fields, in the form of plain text “before image” and “after image” of each field.

TRACKING ATTACHMENT CHANGES

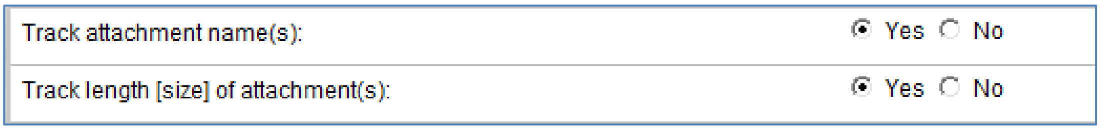
An option is available for the **logging of the names and sizes (lengths) of attachments** before and after a Rich Text field is updated.

In the **Field Tracking Controls** tab of the NotesTracker Configuration document, two fields were added for attachment tracking.

- The **Track attachment name(s)** radio button field (having a default value of “Yes”) determines whether or not attachment tracking is carried out for this database.

- The **Track length [size] of attachment(s)** radio button field (also having a default value of “Yes”) determines whether or not the size of each attachment is recorded and displayed following the attachment's name.

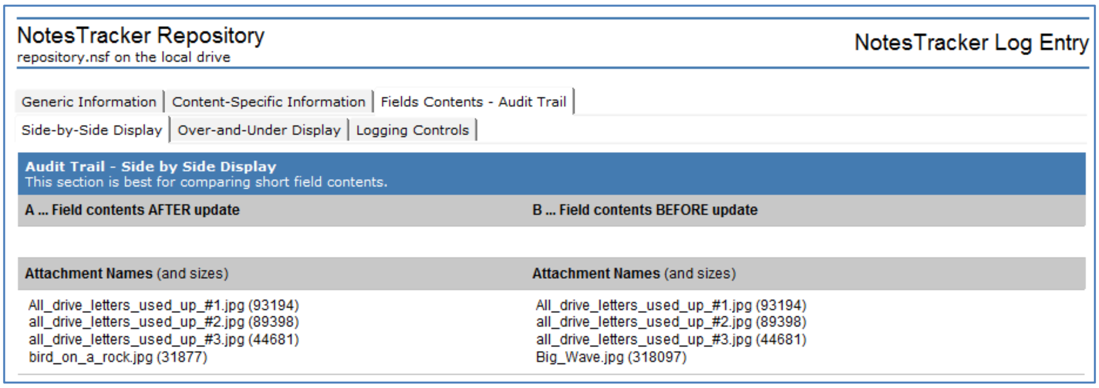
The two fields from the ‘*Field Tracking Controls → Field Tracking Options*’ section of the NotesTracker Configuration document are shown below:



Track attachment name(s): ● Yes ○ No

Track length [size] of attachment(s): ● Yes ○ No

Here is an example of the output produced by this tracking option:



NotesTracker Repository
repository.nsf on the local drive

NotesTracker Log Entry

Generic Information | Content-Specific Information | Fields Contents - Audit Trail |
Side-by-Side Display | Over-and-Under Display | Logging Controls |

Audit Trail - Side by Side Display
This section is best for comparing short field contents.

A ... Field contents AFTER update	B ... Field contents BEFORE update
Attachment Names (and sizes)	**Attachment Names** (and sizes)
All_drive_letters_used_up_#1.jpg (93194) all_drive_letters_used_up_#2.jpg (89398) all_drive_letters_used_up_#3.jpg (44681) bird_on_a_rock.jpg (31877)	All_drive_letters_used_up_#1.jpg (93194) all_drive_letters_used_up_#2.jpg (89398) all_drive_letters_used_up_#3.jpg (44681) Big_Wave.jpg (318097)

In the above example the attachment ‘Big_Wave.jpg’ was removed and ‘bird_on_a_rock.jpg’ was added. No other field editing occurred, so the “Field contents BEFORE update” and “Field contents AFTER update” are empty.

If you need more details about attachments than this, please contact AGECOM to discuss possible additional design features (which could be provided in the form of fee-based consulting services). The e-mail address for our NotesTracker support team is: notestracker@agecom.com.au

NOMINATING FIELDS TO BE SUPPRESSED FROM THE LOGGING OF CHANGES

An important feature of NotesTracker is the ability to specify which fields in a database should **not** be tracked, that is, fields for which the logging of field content changes should be suppressed.

Reasons for wanting to suppress the logging of some fields are discussed further down.

The bottom half of the **Field Tracking Options** tab on the NotesTracker Configuration page has several options that enable you to remove those fields in a database that are of no interest to you as regards their contents. Such fields would include special fields used internally by Notes (such as the so-called $ fields), and fields that are only used for content display (rather than permanent storage in the database) or as hidden work fields. NotesTracker will faithfully record them all unless you specify otherwise.

WILD-CARD FILTERING OF FIELDS

The default is that fields are suppressed if their field names **begin with** "$" and "UsageTrack" (the reason for these two is explained a few paragraphs later). You can also suppress fields having name that **end with, contain,** or **match** specified strings.

These four "filter" fields provide a lot of versatility in the nomination of fields to be suppressed.

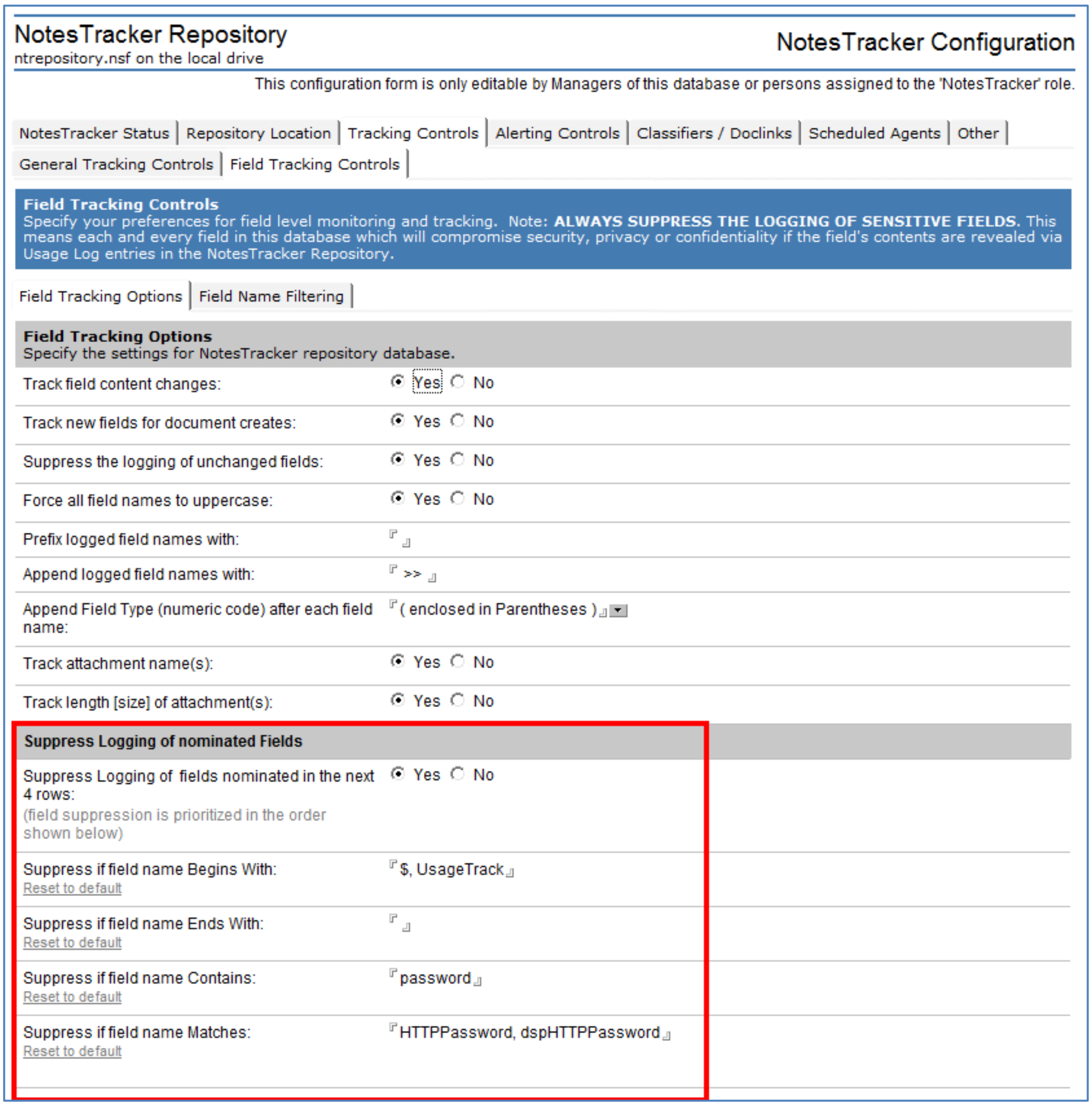
When you save the NotesTracker configuration document, these filter fields are saved too. Therefore if you modify or extend these filters, they are recalled the next time you open the configuration document.

There are some filter strings that apply in a generic fashion to Notes (and NotesTracker), but there will be other filter strings that are unique to each database, so saving them is a handy feature. If you want to go back to scratch, each of the four fields has a button associated with it to restore the NotesTracker defaults for that filter type.

When entering multiple values separate each value with a comma.

It would be a simple matter for your Notes developer to tailor the field names affected by these buttons to better meet the requirements of a given database or of your Notes installation as a whole. Please note we welcome your suggestions for commonly-used field names that we might use behind these buttons in future releases of NotesTracker.

The field filtering section of the NotesTracker Configuration page is highlight in red below in the following diagram:



NotesTracker Repository
ntrepository.nsf on the local drive

NotesTracker Configuration

This configuration form is only editable by Managers of this database or persons assigned to the 'NotesTracker' role.

NotesTracker Status | Repository Location | Tracking Controls | Alerting Controls | Classifiers / Doclinks | Scheduled Agents | Other
General Tracking Controls | Field Tracking Controls

Field Tracking Controls
Specify your preferences for field level monitoring and tracking. Note: **ALWAYS SUPPRESS THE LOGGING OF SENSITIVE FIELDS.** This means each and every field in this database which will compromise security, privacy or confidentiality if the field's contents are revealed via Usage Log entries in the NotesTracker Repository.

Field Tracking Options
Specify the settings for NotesTracker repository database.

Track field content changes:	● Yes ○ No
Track new fields for document creates:	● Yes ○ No
Suppress the logging of unchanged fields:	● Yes ○ No
Force all field names to uppercase:	● Yes ○ No
Prefix logged field names with:	
Append logged field names with:	>>
Append Field Type (numeric code) after each field name:	(enclosed in Parentheses)
Track attachment name(s):	● Yes ○ No
Track length [size] of attachment(s):	● Yes ○ No

Suppress Logging of nominated Fields

Suppress Logging of fields nominated in the next 4 rows: (field suppression is prioritized in the order shown below)	● Yes ○ No
Suppress if field name Begins With: Reset to default	$, UsageTrack
Suppress if field name Ends With: Reset to default	
Suppress if field name Contains: Reset to default	password
Suppress if field name Matches: Reset to default	HTTPPassword, dspHTTPPassword

There are **at least two good reasons** why you would not want to log the before/after images of certain of the fields that were changed in a document edit:

1. The contents may need to be suppressed for reasons of **security, confidentiality or privacy.** Examples of fields in this category are password fields (such as the HTTP Password field in a Person document in the Domino Directory database), personal information (salary, medical, HR, etc), plus such things as board decisions, product strategies, sales results, and mail memo contents. What would be the point of placing such information in highly-restricted databases if logging reveals it?

2. There are probably many **fields of little or no interest for logging purposes**.
One category is fields used internally by Notes, such as the so-called "dollar fields" like $UpdatedBy, $Ref, and $Revisions. As well, there will be many fields in your databases that are not of any significance for content tracking and KM (Knowledge Management) purposes – hidden fields, work fields, computed-for-display fields, and the like.

Since there may be dozens or even hundreds of fields in a form, using the three types of **wild-card specifiers** can greatly reduce the burden for field name elimination. Otherwise, you must list the field names explicitly.

Let's go into a little more detail about the filter strings. Firstly, you specify the field names via simple strings that are not surrounded by quote characters, and the names are not case-sensitive (since the names are all converted to upper case during comparisons). The search for a matching field name proceeds in the following order (stopping after the first match, if any):

1. **Starting string** – such as: **$**, **UsageTrack** (for fields generated by NotesTracker itself), **thread** (in, say, a discussion database), and **temp** or **work** (which you might use to signify temporary work fields).

2. **Ending string** – such as: **disp** or **_disp** (which you might use as a convention for indicating computed- for-display fields).

3. **Embedded string** – such as: **password** or **salary** or **hidden** (a match occurs if such a string is found anywhere in the field name).

4. **Exact match** (do this when the above filters aren't suitable) – such as: **form**, **author**, **httppassword** or **HTTPPassword** (uppercase or lowercase or mixed case makes no difference, except perhaps to legibility).

Since there often are very many fields in a database, NotesTracker provides a means of easily discovering their names so that you can add the names to one or other of the four filter fields just discussed.

Going one step further, how do you determine what are the names of the fields in the current database? The solution is to switch to the adjacent "**Field Name Filtering**" tab. This was designed to generate a list of the names of all fields in the ***forms*** in the database.

Clicking buttons in the top section of this tab generate lists of field names, and another button at the bottom of the tab can be used to remove "noise" or "clutter" in the field name lists so that you can hone in on the field names that are of value for usage tracking purposes.

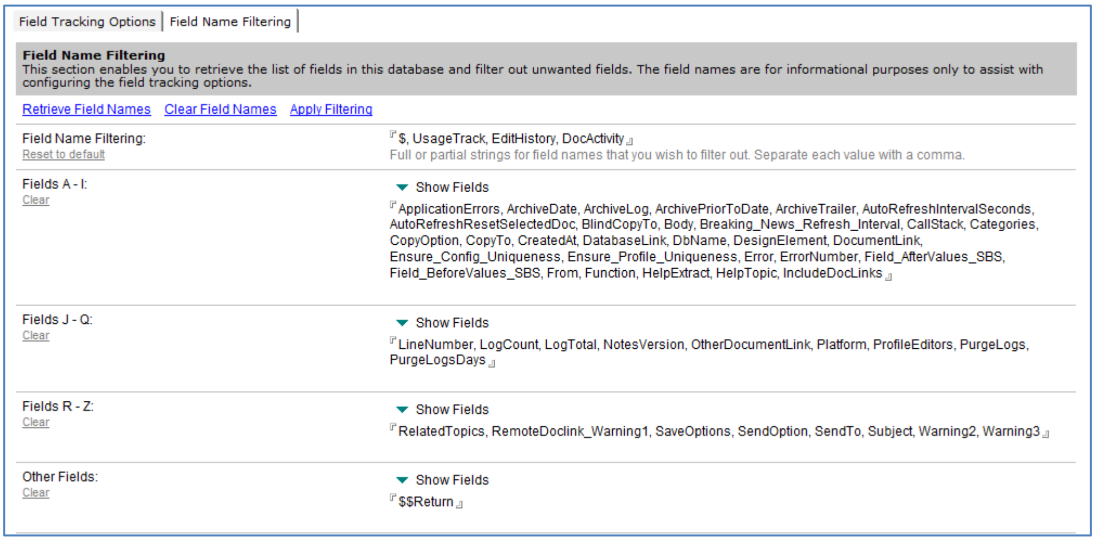
Note: the buttons do not gather the names of fields that are stored in the ***documents*** that are currently stored in the database. A database may contain documents that were not created (or updated) via forms presented to the “front end” (the user interface, either Notes Client or Web browser). Such things as agents and other “back end” processes can store data fields in one or more of the database’s documents, and some or all of this data may be allotted field names that do not exist in the database’s forms. The buttons have been designed to gather field names from all the forms (a rapid process) rather than all of the documents, since this could be a very slow process in a database holding a large number of documents. If there are any such fields that you want to track, it’s a simple matter to type them in manually.

Four buttons are available which break down the field name lists into alphabetical groupings - this is because a single button can run into memory issues due to Notes limitations when a database has many hundreds of fields.

The steps involved are faster to carry out than to describe. They should a matter of seconds, or at most a minute or two, to complete. A brief description follows.

LISTING THE NAMES OF FIELDS IN THE DATABASE, AND FIELD NAME FILTERING

Go to the **Field Name Filtering** tab and open the twisties in the top section labeled. Click the 'Retrieve Field Names' link to retrieve the list of fields from all or selected forms in the application. The field names lists will of course vary from database to database, but you should get a result like this:



Field Name Filtering
This section enables you to retrieve the list of fields in this database and filter out unwanted fields. The field names are for informational purposes only to assist with configuring the field tracking options.

Field Name Filtering: Reset to default	$, UsageTrack, EditHistory, DocActivity Full or partial strings for field names that you wish to filter out. Separate each value with a comma.
Fields A - I: Clear	Show Fields ApplicationErrors, ArchiveDate, ArchiveLog, ArchivePriorToDate, ArchiveTrailer, AutoRefreshIntervalSeconds, AutoRefreshResetSelectedDoc, BlindCopyTo, Body, Breaking_News_Refresh_Interval, CallStack, Categories, CopyOption, CopyTo, CreatedAt, DatabaseLink, DbName, DesignElement, DocumentLink, Ensure_Config_Uniqueness, Ensure_Profile_Uniqueness, Error, ErrorNumber, Field_AfterValues_SBS, Field_BeforeValues_SBS, From, Function, HelpExtract, HelpTopic, IncludeDocLinks
Fields J - Q: Clear	Show Fields LineNumber, LogCount, LogTotal, NotesVersion, OtherDocumentLink, Platform, ProfileEditors, PurgeLogs, PurgeLogsDays
Fields R - Z: Clear	Show Fields RelatedTopics, RemoteDoclink_Warning1, SaveOptions, SendOption, SendTo, Subject, Warning2, Warning3
Other Fields: Clear	Show Fields $$Return

The generation of the field name lists should be very fast, usually a few seconds, but will take a little longer if there are lots of fields in the database.

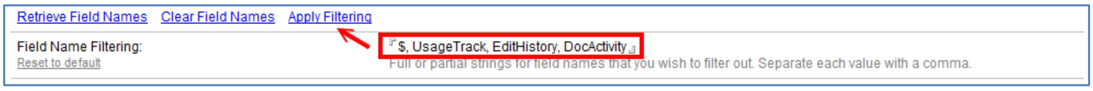
To clear the lists of fields names from each field click the 'Clear Field Names' link at the top of the page.

Filtering Unwanted Field Names

The idea of the Field Tracking Controls tab is to nominate fields that you wish to track. But even a moderately complicated database can contain hundreds of field names, making it rather difficult to "see the wood for the trees" just due to the sheer size of the field name lists.

Coming to your rescue is a simple means of quickly filtering out unwanted field names, via an operation that you can carry out very rapidly either a single time or repetitively until the lists are reduced to manageable proportions.

There is a default string of characters ("$, UsageTrack, EditHistory, DocActivity") to act as a filter. Use the default list as provided or modify it to suit your filtering needs.



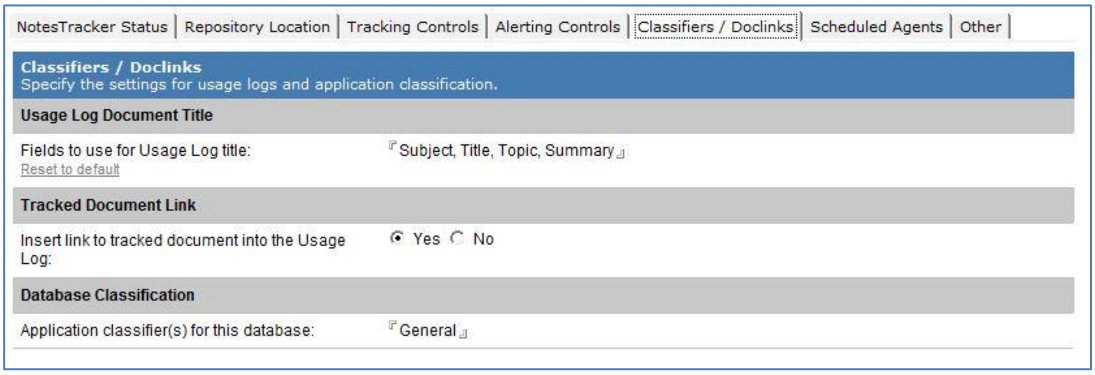
Field Name Filtering: Reset to default	$, UsageTrack, EditHistory, DocActivity Full or partial strings for field names that you wish to filter out. Separate each value with a comma.

Edit the list of filter strings as required (separate each filter value with a comma) for unwanted fields that are common in this particular database then click the 'Apply Filtering' link at the top of the page.

You will now have shorter lists of field names that can be copied to the clipboard and pasted into the various tracking suppression lists under the Field Tracking Options tab.

SPECIFYING A DATABASE'S TITLE, DOCLINK AND APPLICATION CLASSIFICATION

The '**Classifiers / Doclinks**' tab allows you to specify several general-purpose settings:


NotesTracker Status | Repository Location | Tracking Controls | Alerting Controls | Classifiers / Doclinks | Scheduled Agents | Other

Classifiers / Doclinks
Specify the settings for usage logs and application classification.

Usage Log Document Title

Fields to use for Usage Log title: Subject, Title, Topic, Summary
Reset to default

Tracked Document Link

Insert link to tracked document into the Usage Log: ● Yes ○ No

Database Classification

Application classifier(s) for this database: General

These settings apply to every type of action performed against the database, with just one exception, namely that Doclinks are not relevant for document deletions (since you obviously cannot link to a non-existent document).

SETTING UP A MEANINGFUL 'TITLE / SUBJECT / TOPIC' FIELD FOR EACH USAGE LOG ENTRY

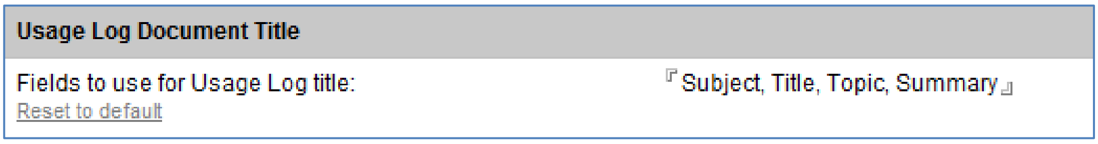
NotesTracker produces “usage log” documents, or entries – one for each operation (Create, Read, Update, Delete, or view switch) in every one of your databases being tracked. These entries are your means of “usage metric analysis”, and the entries are presented as various categorized views in the NotesTracker repository database.

Each row in a view has a particular column that displays a “title” for the document that was logged. The question arises: how do you ensure that there is a meaningful title to display for any given usage log document, so that when you traverse a view it is immediately fairly apparent to you (without your having to open the usage log document) which underlying database document it refers to?

For each individual database being tracked (and its replicas), a list of field names can be set up that will be searched for in the current document being logged and then used as the title of its Usage Log entry.

The list of fields is specified in the ‘Classifiers / Doclinks’ section of the NotesTracker Configuration profile. The default list of field names is: "Subject", "Title", "Topic" and "Summary"

You may alter this list of field names to suit each individual database application.



Usage Log Document Title

Fields to use for Usage Log title: Reset to default	『Subject, Title, Topic, Summary』

For example, you can alter the order in which the field names are listed. Or you can insert/append other field names (such as “Abstract” or “Heading”) that are pertinent to the particular database. This gives a good degree of control over the Usage Log entry’s title, without the need for any programming.

Note: the field names in each document in a database being tracked are searched for in the order specified in this list. The first occurrence in the current document of a field name from this list determines which field’s contents is used as the usage log entry’s “title” for that document.

You can reset the list of names to the default values by clicking the ‘Reset to default’ link underneath the field label.

THE “USAGETRACKING_TITLE” SPECIAL TRACKING FIELD

Normally, as described just above, NotesTracker works its way down the list of field names ("Subject", "Title", "Topic" and "Summary" -- or whatever you supply), and looks in the document being tracked for a matching field name. Upon encountering a match, NotesTracker uses the contents of the matched field as the “title” for the Usage Log document.

But what do you do if a document being tracked has no such suitable title field (in its form design)?

The NotesTracker solution is for your Notes developer to add a field called “**UsageTracking_Title**” to the form design, and populate it with a character string that is appropriate for the document. The contents of the **UsageTracking_Title** field are used as the usage log entry’s “title” or “Subject” or “Topic” or “Summary” for that document – taking precedence over the contents of any of the fields in the field names list.

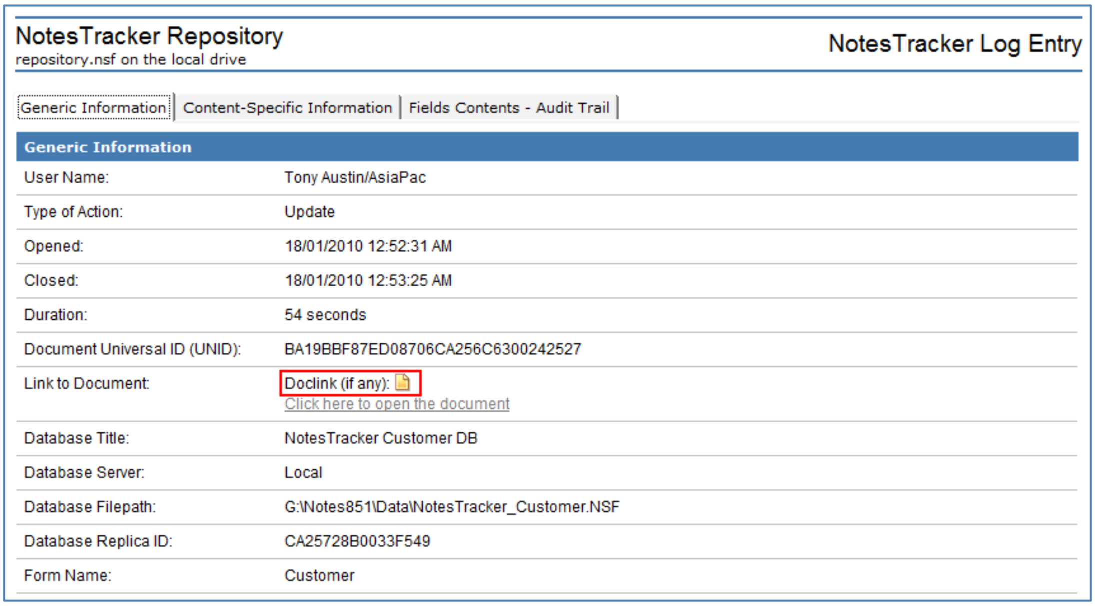
There is a detailed discussion of this field in the NotesTracker Developer Guide, since it is an aspect of usage logging that your developer must set up.

SUPPRESSING THE LOGGING OF DOCUMENT LINKS

NotesTracker can optionally store in the Usage Log document a **Document Link –** often abbreviated to “**DocLink**” or “Doclink” -- that points to current Notes document for each Read, Create or Update.

Note: storing of a DocLink is never done for a document deletion event. Obviously, unless the deletion request fails, no document will be left in the database for such a link to be valid. Also, DocLinks are not applicable to View Open events, but they do apply for document pastes and mail-ins.

The DocLink makes it easy for you to open the original document when you are examining a particular Usage Log entry, simply by clicking on the DocLink icon.

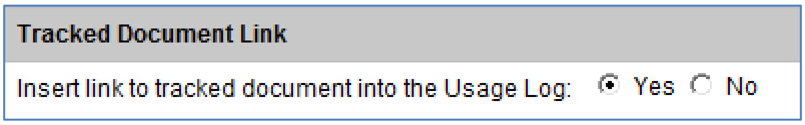


NotesTracker Repository
repository.nsf on the local drive

NotesTracker Log Entry

Generic Information	
User Name:	Tony Austin/AsiaPac
Type of Action:	Update
Opened:	18/01/2010 12:52:31 AM
Closed:	18/01/2010 12:53:25 AM
Duration:	54 seconds
Document Universal ID (UNID):	BA19BBF87ED08706CA256C6300242527
Link to Document:	Doclink (if any): 📄 Click here to open the document
Database Title:	NotesTracker Customer DB
Database Server:	Local
Database Filepath:	G:\Notes851\Data\NotesTracker_Customer.NSF
Database Replica ID:	CA25728B0033F549
Form Name:	Customer

The following setting in the ‘Classifiers / Doclinks’ section of the NotesTracker Configuration page enables you to specify whether or not to store any DocLinks for the current database:



Tracked Document Link

Insert link to tracked document into the Usage Log: ● Yes ○ No

Selecting “No” perhaps will save a tiny bit of system resources – processor time, disk space – but probably not much.

Generally it's probably wise not to deselect this option unless you're quite sure that you don't mind losing the convenience of having these links in the Usage Log documents. Furthermore, Doclinks can play a part in the dynamic opening of documents for portal pages, discussed elsewhere in this guide, and this functionality will be lost of Doclinks are suppressed.

In particular, the existence of a DocLink in each Usage Log document is essential for the nifty “Breaking News” technique (described in the NotesTracker Developer Guide) to operate properly. For this technique to succeed, the DocLink must not be suppressed. Otherwise, when the Usage Log document is opened via the Breaking News view, there is no DocLink present that will allow automatic displaying of the underlying Notes document. The end user will not want to see the Usage Log document (the technical content of which is of absolutely no interest to him/her). For this “Breaking News” approach, you only ever want the user to see *the original Notes document* (the one that was originally tracked and which holds the interesting content).

Note: you might be interested in the article **Managing Lotus Notes DocLinks with LotusScript** at:

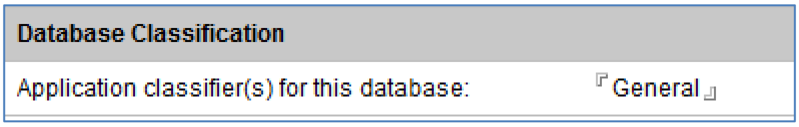
https://www.agecom.com.au/support/agecomkb.nsf/0/80c21105bbe64e3eca257d76006e2b4e?OpenDocument

which shows code for two agents, the first of which that builds a list of all of the DocLinks in a database. The second agent is then run to determine whether or not the target document behind each DocLink still exists.

SPECIFY A DATABASE'S APPLICATION CLASSIFICATION

A feature is available to specify "classifications" or "categories" for each database being tracked, with the purpose of enabling extended analysis and reporting in the NotesTracker Repository.

The classification value is set in the 'Classifiers / Doclinks' section of the NotesTracker Configuration page and defaults to "**General**.



Database Classification

Application classifier(s) for this database: 『General』

Put one or more entries in the classification field to indicate the database's nature. Enter the classifiers as a string of characters, using commas as separators between the classifier entries. There is no need to enter spaces between entries. Leading and trailing spaces in each entry will be trimmed out.

These entries will be featured as category heading in Usage Log views, so it pays to check their spelling

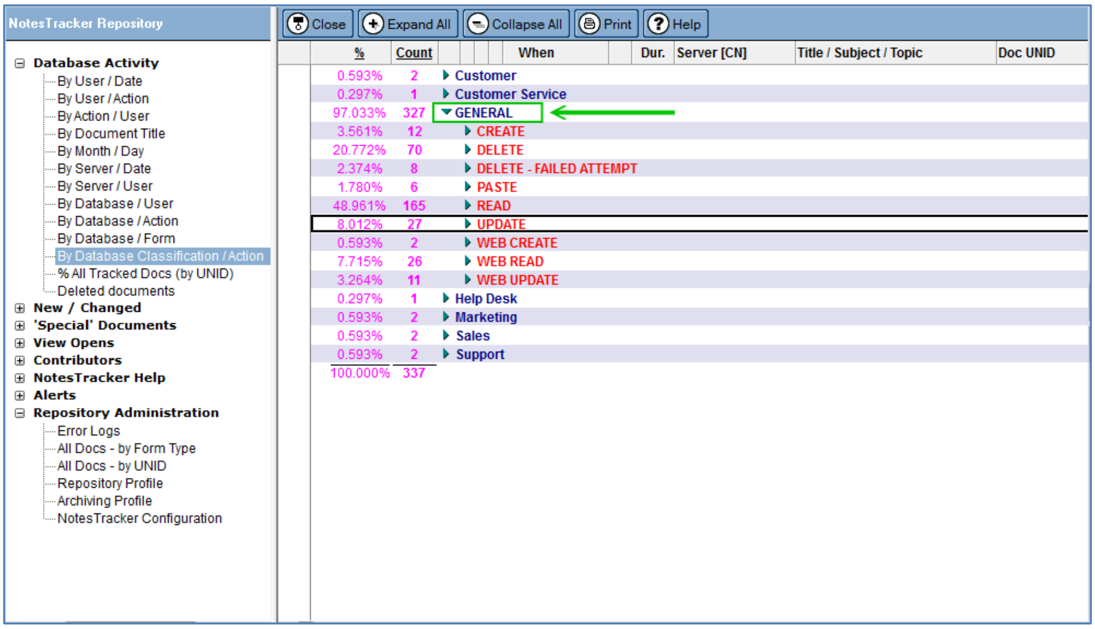
You should discuss this option with the person(s) who "own" or "sponsor" the current database application, or group of related applications, and/or who will be spending most time analyzing the Usage Log.

Refer to the section *What to Track? Individual Notes Databases or Sets of Databases?* for an illustration of some such groups of applications.

Enter an application classification value, or multiple values, whatever is most pertinent for the current database. Thus, for a CRM database you might choose: **Marketing, Sales, Customers, Project Tracking, Appointments.**

If you leave this field blank – and for compatibility with Usage Log entries created in earlier versions of NotesTracker (where this Classifier field was not generated) - the special value of "GENERAL" appears in the "**By Database Classification / Action**" view in the NotesTracker Repository database.

The following example shows several classifications (Customer Service, Help Desk, Marketing, Sales) as well as the GENERAL value:



%	Count		When	Dur.	Server [CN]	Title / Subject / Topic	Doc UNID
0.593%	2	▸ Customer					
0.297%	1	▸ Customer Service					
97.033%	327	▾ GENERAL					
3.561%	12	▸ CREATE					
20.772%	70	▸ DELETE					
2.374%	8	▸ DELETE - FAILED ATTEMPT					
1.780%	6	▸ PASTE					
48.961%	165	▸ READ					
8.012%	27	▸ UPDATE					
0.593%	2	▸ WEB CREATE					
7.715%	26	▸ WEB READ					
3.264%	11	▸ WEB UPDATE					
0.297%	1	▸ Help Desk					
0.593%	2	▸ Marketing					
0.593%	2	▸ Sales					
0.593%	2	▸ Support					
100.000%	337						

In summary, this feature offers you a new **business-centric perspective for analyzing database usage patterns and trends** over the months and years.

PROACTIVITY VIA EMAIL ALERTING AND “SPECIAL” DOCUMENTS

A tracking feature is available to monitor “**special documents**” - specific documents in a database being of extra interest or in some way more important than other documents in the database.

In the NotesTracker Repository, the special documents are given **their own separate Usage Log views**, so that they stand out as soon as they appear in the Repository.

And even better if, instead of just passively keeping a watch on the repository, you can specify that you want to be proactively alerted via e-mail whenever one of these special documents is touched in some way (created, updated, deleted, etc). Going one step further, you could even try using a third-party Domino add-on that detects the arrival of each such e-mail message and immediately triggers an SMS or pager alert.

Two features available in NotesTracker can handle these requirements, with minimal developer involvement:

- **SPECIAL DOCUMENTS:**
  The ability to easily nominate any document in a database as a “special document”

- **EMAIL ALERTING:**
  The ability to nominate some or all of the action types to automatically send an e-mail message to delegated recipient(s).

  The document actions that you can select (via checkbox) are any combination of “the usual suspects” … Create, Read, Update, Delete, Paste and Mail-in.

  For **updates** you can go one step further and select a list of **specific fields** that, when updated, will generate or “trigger” an e-mail alert (hence the selected fields are referred to as “trigger fields”). For **special documents**, you can specify the e-mail alert to be generated **for any action** or just for **the selected actions**.

Let’s consider two examples – and there would be many others – of how e-mail alerting for “special documents” would come in handy.

Firstly, in any moderate-to-large sized organization, Domino administrators are likely to be more focused on certain strategic Domino servers than on the others. They are in one way or another critical to high operational availability and service levels: typically hub servers involved with replication, mail routing, Web serving, etc (plus, almost certainly, any server dedicated to the top executives in the organization – after all, “You’ve got to keep the boss happy!”).

The other Domino servers would be regarded as being, relatively speaking, of somewhat lesser importance. It follows that the administrators would be more interested in changes happening to or deletion of certain Server documents than others. So they would designate the Server documents for these strategic servers as “special documents” (probably deselecting document Read operations as being of little interest). Perhaps they might even nominate certain Server Document fields to act as triggers: security- related fields (such as Full Access administrators), programmability restrictions (Run unrestricted methods, etc), performance-related fields (the likes of Maximum number of threads), and so on.

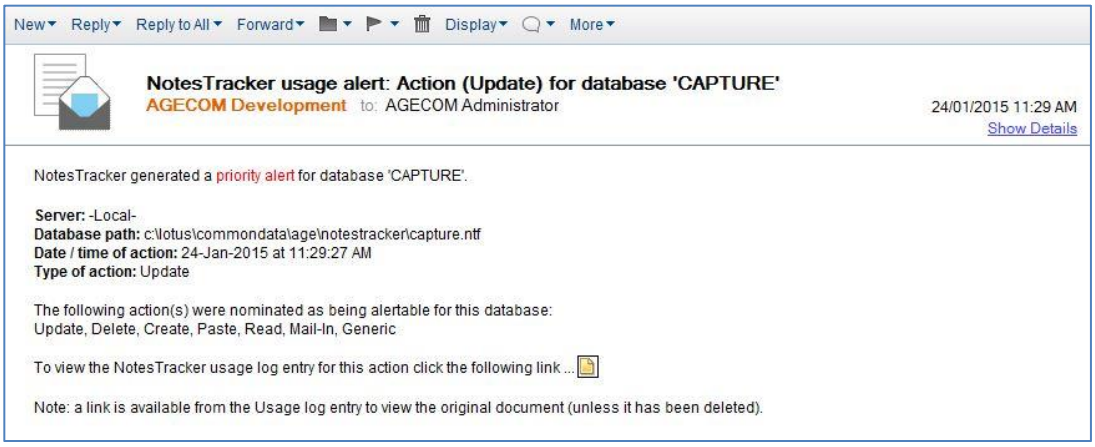
A second example is a Customer Relationship Management (CRM) application where sales people would benefit from being proactively alerted via e-mail as soon as there is activity against specific customers or prospects: for example, a Sales Call Report document being created or a customer’s address and telephone number being changed.

For CRM applications like this, usually there will be certain customers (or prospects, or business partners) who are of considerable importance to the organization's current sales and marketing efforts. Therefore it would be extremely beneficial to put such customers onto a watch list, and with NotesTracker V5 you would do this by nominating as "special documents" some specific documents in the CRM database relating to these customers. After that, not only would a passive watch on these customers (via the continually-updated NotesTracker usage repository) but also you could go to the next step and have e-mail alerts generated as soon as the special documents are touched. Then later on, when the marketing or sales focus moves to other customers, you could easily and quickly change which of the documents in the database are nominated as being "special" and thereby easily adapt to a rapidly changing marketing and sales environment.

EXAMPLE OF NOTESTRACKER EMAIL ALERT MESSAGES

Each addressee who is specified to receive email alerts for a given database will get an incoming mail message in one of two layouts.

For specified event(s) happening to general documents (those not marked as "special documents"):



NotesTracker usage alert: Action (Update) for database 'CAPTURE'
AGECOM Development to: AGECOM Administrator
24/01/2015 11:29 AM

NotesTracker generated a priority alert for database 'CAPTURE'.

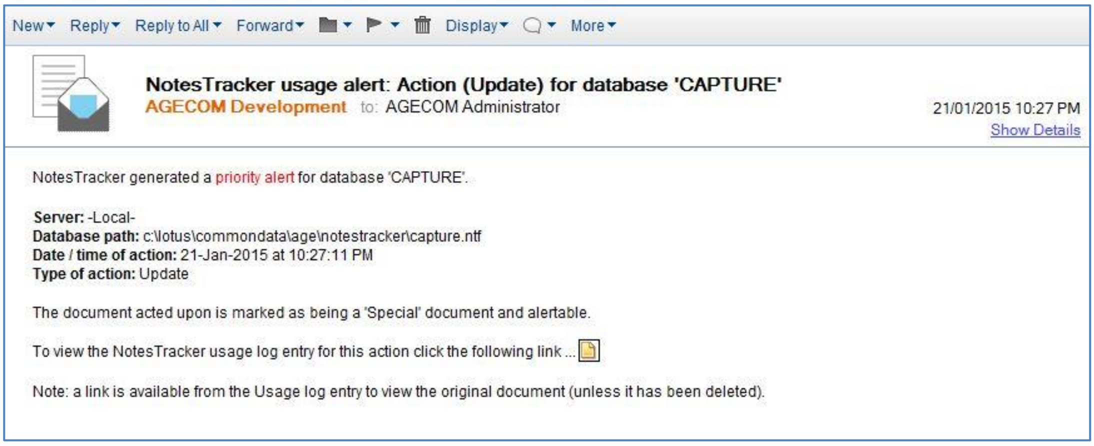
Server: -Local-
Database path: c:\lotus\commondata\age\notestracker\capture.ntf
Date / time of action: 24-Jan-2015 at 11:29:27 AM
Type of action: Update

The following action(s) were nominated as being alertable for this database:
Update, Delete, Create, Paste, Read, Mail-In, Generic

To view the NotesTracker usage log entry for this action click the following link ...

Note: a link is available from the Usage log entry to view the original document (unless it has been deleted).

For specified action types (or all action types, if so selected), when the document is a "special" document:



NotesTracker usage alert: Action (Update) for database 'CAPTURE'
AGECOM Development to: AGECOM Administrator
21/01/2015 10:27 PM

NotesTracker generated a priority alert for database 'CAPTURE'.

Server: -Local-
Database path: c:\lotus\commondata\age\notestracker\capture.ntf
Date / time of action: 21-Jan-2015 at 10:27:11 PM
Type of action: Update

The document acted upon is marked as being a 'Special' document and alertable.

To view the NotesTracker usage log entry for this action click the following link ...

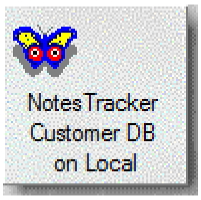
Note: a link is available from the Usage log entry to view the original document (unless it has been deleted).

THE "USAGETRACKING_SPECIALDOC" SPECIAL TRACKING FIELD

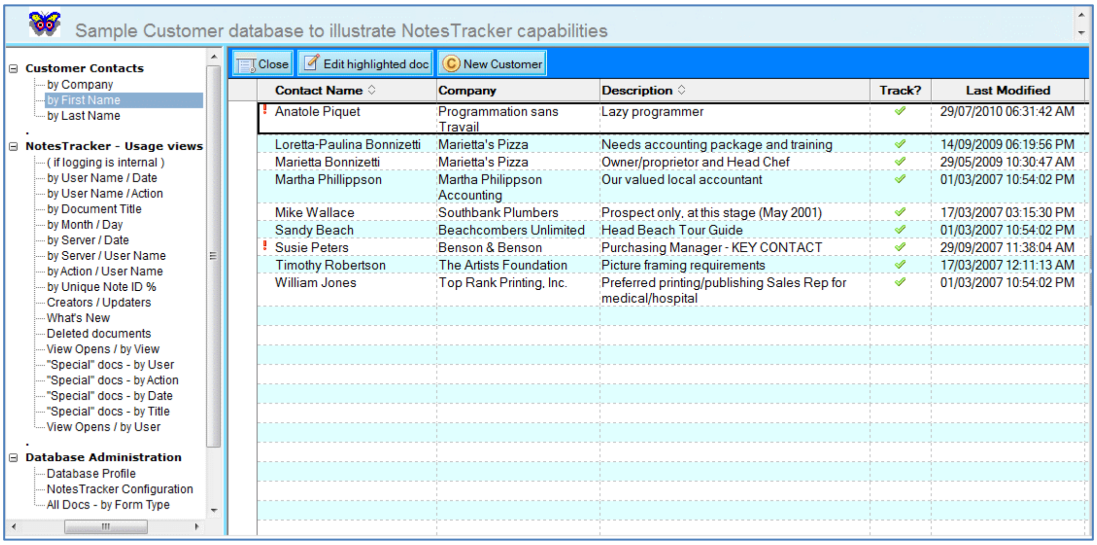
In the NotesTracker Developer Guide, it is explained how, with some quick and simple database design changes, your Notes developer can:

- Set up a special NotesTracker plain text field, called "**UsageTracking_SpecialDoc**".
- Arrange for that field to be populated (either user edited, or under program control).
- Ensure that the field can be used to generate e-mail alerts
- Adjust how the field affects the Usage Log views in the NotesTracker Repository database for maximum impact

But let's assume that you have a database with a form already containing the UsageTracking_SpecialDoc text field. A good example of this is the example **NotesTracker Customer Database** that is part of the NotesTracker distribution package:



This database has a design with the only intent being to clearly illustrate a range of NotesTracker capabilities, so its design is very straightforward: a single Customer form a few views (Customers by Company Name, by contact First Name, by contact Last Name) as well as a set of NotesTracker usage log views and a few administration/control views.


Sample Customer database to illustrate NotesTracker capabilities
Close
Edit highlighted doc
New Customer

Customer Contacts
- by Company
- by First Name
- by Last Name

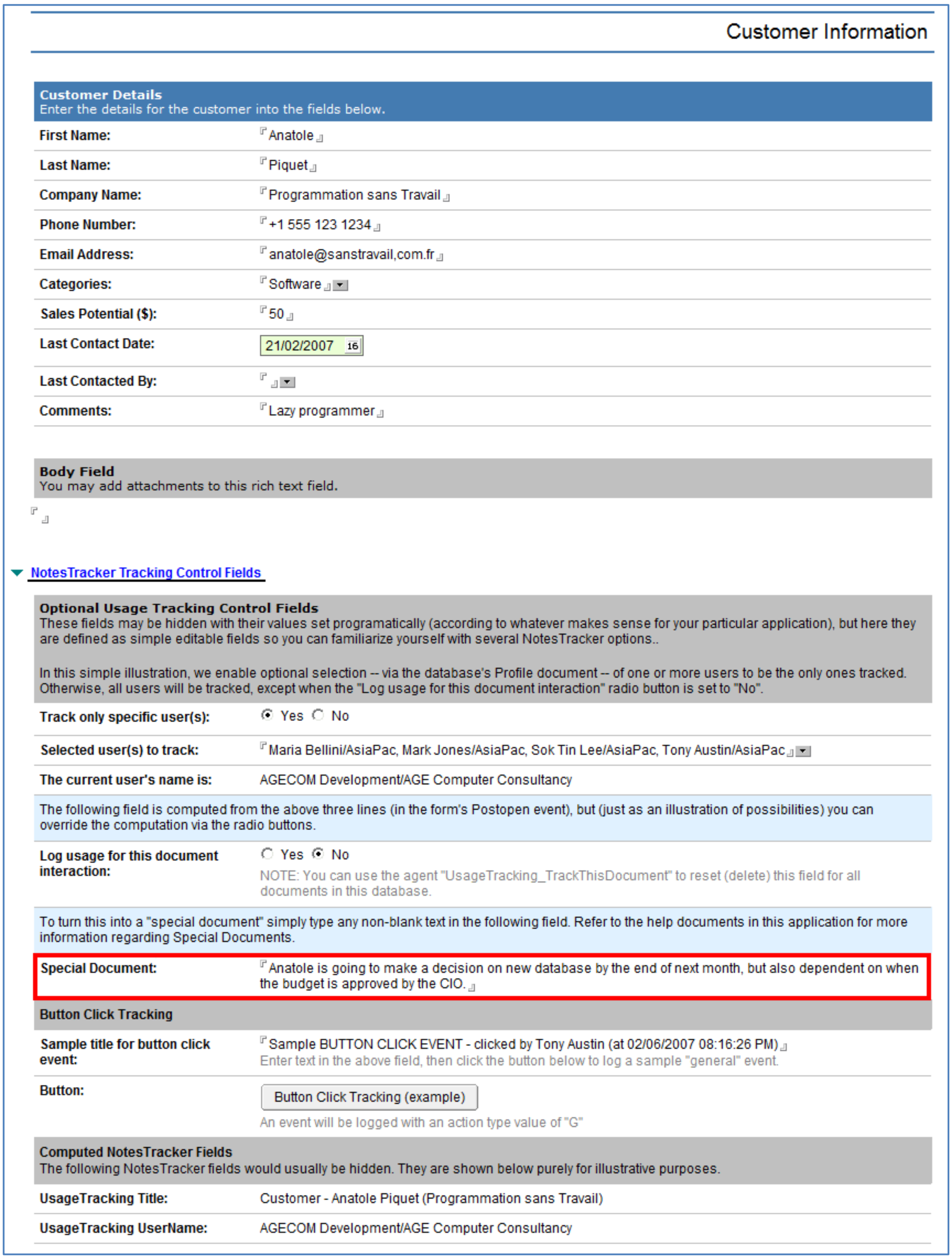
NotesTracker - Usage views
- (if logging is internal)
- by User Name / Date
- by User Name / Action
- by Document Title
- by Month / Day
- by Server / Date
- by Server / User Name
- by Action / User Name
- by Unique Note ID %
- Creators / Updaters
- What's New
- Deleted documents
- View Opens / by View
- "Special" docs - by User
- "Special" docs - by Action
- "Special" docs - by Date
- "Special" docs - by Title
- View Opens / by User

Database Administration
- Database Profile
- NotesTracker Configuration
- All Docs - by Form Type

	Contact Name	Company	Description	Track?	Last Modified
!	Anatole Piquet	Programmation sans Travail	Lazy programmer	✓	29/07/2010 06:31:42 AM
	Loretta-Paulina Bonnizetti	Marietta's Pizza	Needs accounting package and training	✓	14/09/2009 06:19:56 PM
	Marietta Bonnizetti	Marietta's Pizza	Owner/proprietor and Head Chef	✓	29/05/2009 10:30:47 AM
	Martha Phillippson	Martha Philippson Accounting	Our valued local accountant	✓	01/03/2007 10:54:02 PM
	Mike Wallace	Southbank Plumbers	Prospect only, at this stage (May 2001)	✓	17/03/2007 03:15:30 PM
	Sandy Beach	Beachcombers Unlimited	Head Beach Tour Guide	✓	01/03/2007 10:54:02 PM
!	Susie Peters	Benson & Benson	Purchasing Manager - KEY CONTACT	✓	29/09/2007 11:38:04 AM
	Timothy Robertson	The Artists Foundation	Picture framing requirements	✓	17/03/2007 12:11:13 AM
	William Jones	Top Rank Printing, Inc.	Preferred printing/publishing Sales Rep for medical/hospital	✓	01/03/2007 10:54:02 PM

TIP: in the NotesTracker Repository, the convention is used that "special" documents are highlighted with a **red exclamation mark** and that this is placed in the leftmost column (as demonstrated in the above illustration). This is analogous to the way that Lotus Notes mail uses a red exclamation mark to flag high-priority mail messages. It is recommended that, as in this Customer DB example, you **follow this convention in your own databases**.

Here is an example of the simple Customer form, with the **UsageTracking_SpecialDoc** special field being the one highlighted with a red border near the bottom. Of course, users of the database have no inkling of this field's significance. To them it's just another ordinary text field on the form:



Customer Information

Customer Details
Enter the details for the customer into the fields below.

First Name:	Anatole
Last Name:	Piquet
Company Name:	Programmation sans Travail
Phone Number:	+1 555 123 1234
Email Address:	anatole@sanstravail,com.fr
Categories:	Software ▾
Sales Potential ($):	50
Last Contact Date:	21/02/2007
Last Contacted By:	▾
Comments:	Lazy programmer

Body Field
You may add attachments to this rich text field.

▾ NotesTracker Tracking Control Fields

Optional Usage Tracking Control Fields
These fields may be hidden with their values set programatically (according to whatever makes sense for your particular application), but here they are defined as simple editable fields so you can familiarize yourself with several NotesTracker options..

In this simple illustration, we enable optional selection -- via the database's Profile document -- of one or more users to be the only ones tracked. Otherwise, all users will be tracked, except when the "Log usage for this document interaction" radio button is set to "No".

Track only specific user(s):	● Yes ○ No
Selected user(s) to track:	Maria Bellini/AsiaPac, Mark Jones/AsiaPac, Sok Tin Lee/AsiaPac, Tony Austin/AsiaPac ▾
The current user's name is:	AGECOM Development/AGE Computer Consultancy

The following field is computed from the above three lines (in the form's Postopen event), but (just as an illustration of possibilities) you can override the computation via the radio buttons.

Log usage for this document interaction:	○ Yes ● No NOTE: You can use the agent "UsageTracking_TrackThisDocument" to reset (delete) this field for all documents in this database.

To turn this into a "special document" simply type any non-blank text in the following field. Refer to the help documents in this application for more information regarding Special Documents.

Special Document:	Anatole is going to make a decision on new database by the end of next month, but also dependent on when the budget is approved by the CIO.

Button Click Tracking

Sample title for button click event:	Sample BUTTON CLICK EVENT - clicked by Tony Austin (at 02/06/2007 08:16:26 PM) Enter text in the above field, then click the button below to log a sample "general" event.
Button:	Button Click Tracking (example) An event will be logged with an action type value of "G"

Computed NotesTracker Fields
The following NotesTracker fields would usually be hidden. They are shown below purely for illustrative purposes.

UsageTracking Title:	Customer - Anatole Piquet (Programmation sans Travail)
UsageTracking UserName:	AGECOM Development/AGE Computer Consultancy

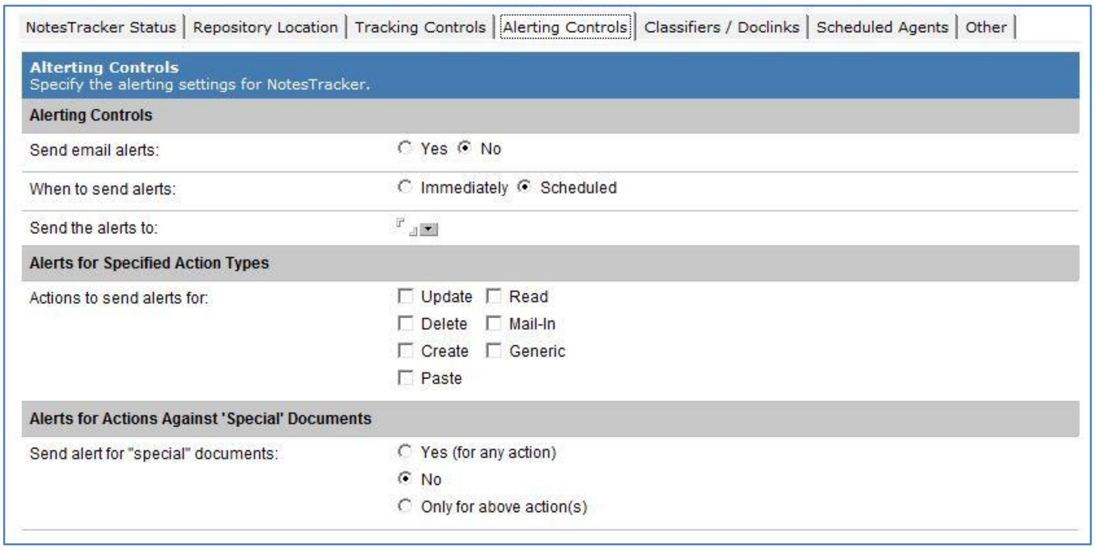
As soon as a user enters some text – and it can be *any* text – into this field, then as far as NotesTracker is concerned this is a “special document” and that’s all there is to it! (it is also possible for your Notes developer to make this a non-editable field, and even a hidden field, the content of which depends on either some other field(s) that the user has populated, on some algorithm relationship between other fields in the form or read in from elsewhere, or a combination of the two. This is discussed in the NotesTracker Developer Guide).

As far at the database administrator is concerned, it is not the field itself that matters, but how it affects the NotesTracker Configuration document, so let’s now switch across to this.

ALERTING CONTROLS

It is via the Alerting Controls tab of the NotesTracker Configuration page that you control whether, to whom and under what circumstances e-mail alerts should be sent out by during usage tracking.

Some options are hidden at certain times. For example, the selection of “trigger fields” only appears for field updates; that is, if you select the Update action type via the check box in the central section of this tab.



Alterting Controls
Specify the alerting settings for NotesTracker.

Alerting Controls

Send email alerts:	○ Yes ● No
When to send alerts:	○ Immediately ● Scheduled
Send the alerts to:	

Alerts for Specified Action Types

Actions to send alerts for:	□ Update □ Read □ Delete □ Mail-In □ Create □ Generic □ Paste

Alerts for Actions Against 'Special' Documents

Send alert for "special" documents:	○ Yes (for any action) ● No ○ Only for above action(s)

ALERTS FOR SPECIFIED ACTION TYPES

Generally you will want to keep alert e-mail traffic within reasonable bounds, both for technical reasons (network traffic, mail server overheads) and so that you don't get a flood of alerts that lead to alerting to lose its effectiveness. Therefore you can specify which of action type(s) can generate alerts for the current database.

SELECTING A SUBSET OF ACTION TYPES

Usually you can switch of alerting for document Reads (which, as stated elsewhere, typically make up 80 to 90 percent of database actions). There may be certain situations where alerting for Reads has value, however you should weigh up the benefits against the disadvantages. Alerts for document Pastes and Mail-ins might be useful, but often won't be. It is most likely that alerts will useful for document Creates, Deletes and Updates.

For Updates, you can go one step further by nominating a subset of the database's fields that can generate alerts if modified. If you are tracking the Domino Directory, say, then you might be especially interested in changes to fields in the Server Document such as "Full Access administrators" (field name "FullAdmin") and "Maximum number of threads" (field name "AdminPMaxThreads"). Or in a Travel Request application, the travel booking department might want to be alerted if an employee makes changes to fields such as "DepartureDate" and "ArrivalDate" (and probably one or two others).

TRIGGER FIELDS

Such fields are termed "trigger fields" and you specify "**Yes**" for the "**Only if nominated "trigger" fields change**" option. Only then does the "**Trigger field name(s)**" field become visible, and it is here that you list the names of the desired trigger fields.

TIP: to get spelling for the trigger field names, use the "**Field Name Filtering**" tab (under the "Field Tracking Controls**"** tab), described elsewhere in this guide. Copy-and-paste the field names from there to avoid transcription errors.

ALERTS FOR ACTIONS AGAINST "SPECIAL" DOCUMENTS

If you do not want e-mail alerts to be sent out for "special documents", you would select the "**No**" radio button in the "**Send alerts for "special" documents**" area of this tab.

If you want e-mail alerts always to be sent out for special documents (regardless of the action type), you would select the "**Yes (for any action)**" radio button.

Otherwise you would select the "**Only for the above actions(s)**" radio button (and in the above example, e-mail alerts would only be sent out for document creates, updates and deletes).

SPECIFYING WHEN E-MAIL ALERTS ARE TO BE SENT

The option “**When to send the e-mail alert**” allows you to specify whether:

- You want the alerts to be sent “immediately” (that is, synchronously). The alert mail is sent when the document is closed. This is a dependable event in the Notes Client environment, perhaps less so in the Web browser environment (if a user simply closes the browser, say, rather than clicking on the Submit button to update the document).

- You want the e-mails to be sent later (asynchronously), via a scheduled agent in the NotesTracker Repository database. This will send out the e-mail alerts in batches, which may be a better performance option but you should carry out tests to determine whether or not this is so for your Domino environment. The schedule period specified for this agent determines how soon after the tracked event the alerts are sent out.)

IMMEDIATE VERSUS SCHEDULED SENDING OF E-MAIL ALERTS

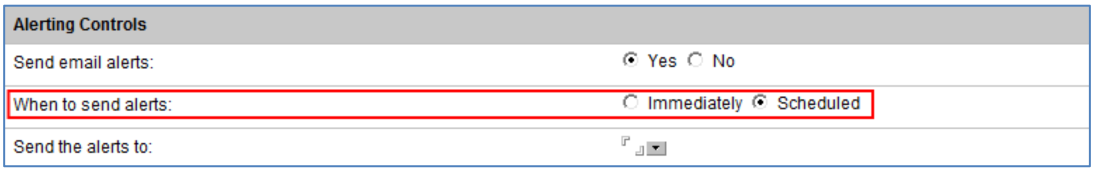
The sending of e-mails places a load on your Domino servers and communications network. Therefore **you should be cautious about how many NotesTracker e-mail alerts get generated.**

Alerts can be generated for “special documents” and/or for regular document events (Creates, Reads, Updates, Deletes, Pastes).

Performance would be affected especially for Reads, which occur very frequently and typically make up some 80 to 90 percent of all database activity.

By all means make use of the alerting capabilities of NotesTracker, that’s why they were added to NotesTracker. Just be sure to use alerting judiciously, striking a balance between their business value and the additional load they place on system resources.

If you do not need alerts to be delivered “synchronously” (at the same time as Usage Log entries being generated), you can change the e-mail delivery option from “**Immediate**” to “**Scheduled**” in the Alerting Controls section of the NotesTracker Configuration page:



Alerting Controls

Send email alerts:	● Yes ○ No
When to send alerts:	○ Immediately ● Scheduled
Send the alerts to:	

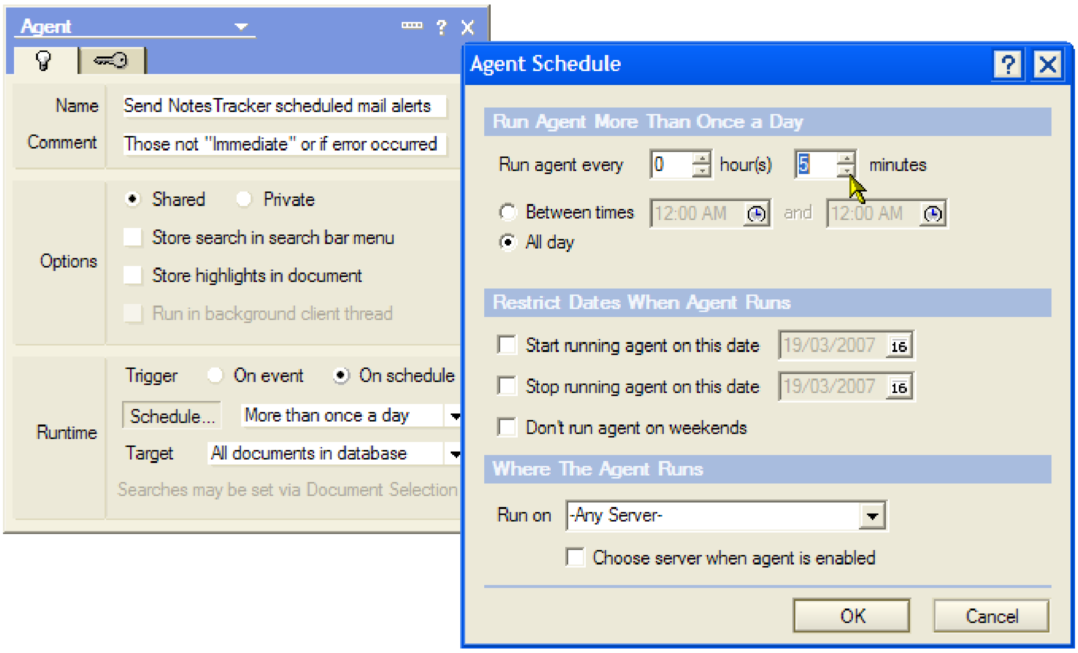
The value “Scheduled” will be stored in the UsageLog document, and e-mail alerts will not be sent synchronously for the current database. An agent in the NotesTracker Repository (or one added to the current database, if usage logging is “internal”) would then send the e-mail alerts at a scheduled frequency. This frequency would be adjusted by you to best balance your Domino server workload against the desired promptness of alert message delivery.

Whether it is better to send alerts synchronously or via agents (asynchronously) will be dependent on your Domino infrastructure and on the nature plus usage patterns of each of your application databases. Only general guidance can be given here. Programming considerations for the immediate versus scheduled sending of NotesTracker e-mail alerts can be found in the NotesTracker Developer Guide.

Let’s look into these matters a little more.

CONSIDERATIONS FOR SCHEDULED ALERTING

The mail is sent out by a scheduled agent named "Send NotesTracker scheduled mail alerts" and, as always, it is incumbent upon the Domino Administrator to set the appropriate schedule options such an agent.



Agent

Name: Send NotesTracker scheduled mail alerts

Comment: Those not "Immediate" or if error occurred

Options: ● Shared ○ Private

□ Store search in search bar menu

□ Store highlights in document

□ Run in background client thread

Runtime: Trigger ○ On event ● On schedule

Schedule... More than once a day

Target: All documents in database

Searches may be set via Document Selection

Agent Schedule

Run Agent More Than Once a Day

Run agent every 0 hour(s) 5 minutes

○ Between times 12:00 AM and 12:00 AM

● All day

Restrict Dates When Agent Runs

□ Start running agent on this date 19/03/2007

□ Stop running agent on this date 19/03/2007

□ Don't run agent on weekends

Where The Agent Runs

Run on -Any Server-

□ Choose server when agent is enabled

Experiment with the scheduling frequency (the "Run agent every xx hour(s) xx minutes" fields) so as to strike the optimum balance between alerting effectiveness and Domino server workload. For testing purposes, the minimum value of 5 minutes gives good turnaround, but this value is almost certainly far too low for your production environment.

ALERTING EFFECTIVENESS, AND GETTING THE BALANCE RIGHT

Think about it. If you select "Immediate" alerting, the Notes Client workstation's resources perform most of the work, and the alert message is sent within a second or two. (If you are using a Web browser, the work is performed on the Domino server, but the message still goes out in a second or two.) Will the recipient(s) notice the incoming alert message immediately? They might, but in many cases probably not until some minutes or even hours later.

Maybe, then, setting the agent's schedule frequency in hours rather than minutes is adequate.

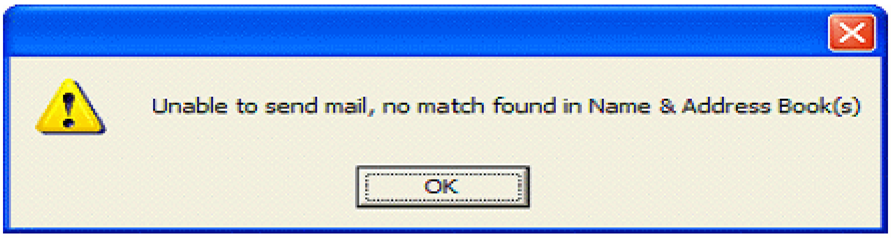
ALERT MAIL RECIPIENTS, AND TESTING OF ALERT SCHEDULING

If you indicate that e-mail alerts are to be sent out, then you must select (via the “**Send the alerts to**” address dialog) the valid Notes Mail address of at least one mail recipient.

It is essential that you carefully test to ensure that – at the time that the alert is going to be sent – the recipient(s) have valid entries in the Domino Directory or other address book(s).

TIP: First carry out your tests using the “Immediate” option. If you intend to use the “Scheduled” option, then carry out the same tests and carefully examine the Domino console log for mail addressing errors.

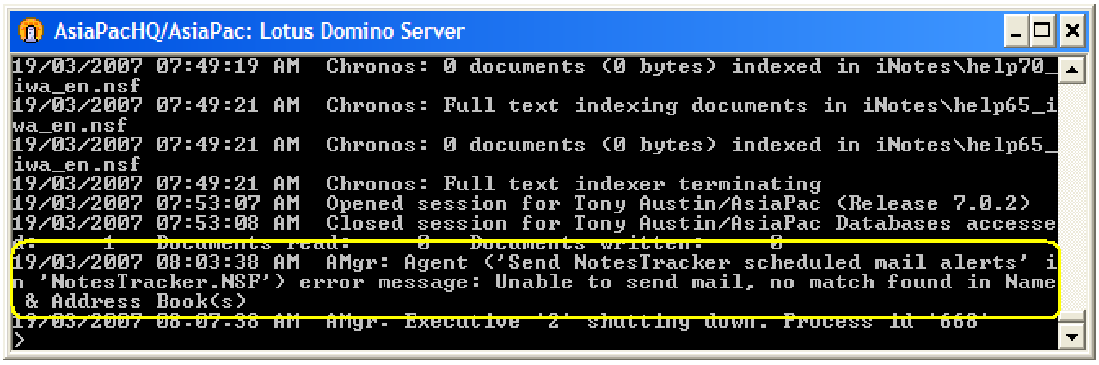
It is a very good idea to use deliberately an invalid e-mail address (for example, “**No Body/There**”) to force an error in both the Immediate and Scheduled alerting situations. In the former, as soon as you go to close the current document you get a mail addressing “**no match found**” error dialog box, like this:



Unable to send mail, no match found in Name & Address Book(s)

In this case, it is quite easy to work back and determine any recipient address that is incorrectly specified. (For example, correct the “No Body/There” error and the dialog box should not appear.)

However, as with most server-based agents, the invalid e-mail address presents a less direct error message on the Domino console (repeated, of course, at every schedule period):



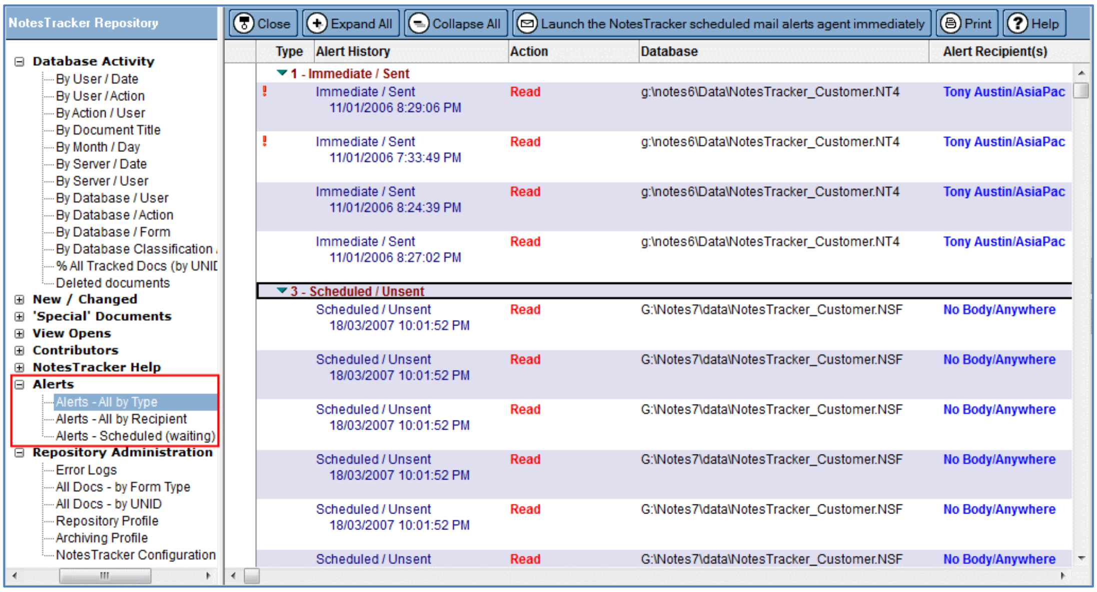
```
19/03/2007 07:49:19 AM  Chronos: 0 documents (0 bytes) indexed in iNotes\help70_
iwa_en.nsf
19/03/2007 07:49:21 AM  Chronos: Full text indexing documents in iNotes\help65_i
wa_en.nsf
19/03/2007 07:49:21 AM  Chronos: 0 documents (0 bytes) indexed in iNotes\help65_
iwa_en.nsf
19/03/2007 07:49:21 AM  Chronos: Full text indexer terminating
19/03/2007 07:53:07 AM  Opened session for Tony Austin/AsiaPac (Release 7.0.2)
19/03/2007 07:53:08 AM  Closed session for Tony Austin/AsiaPac Databases accesse
d:      1  Documents read:      0  Documents written:      0
19/03/2007 08:03:38 AM  AMgr: Agent ('Send NotesTracker scheduled mail alerts' i
n 'NotesTracker.NSF') error message: Unable to send mail, no match found in Name
 & Address Book(s)
19/03/2007 08:07:38 AM  AMgr: Executive '2' shutting down. Process id '668'
>
```

As usual, the error message will get buried in amongst all the other console messages, but a t least you have the agent name and database name as a starting point for your sleuthing work.

MANAGING AND CANCELLING SCHEDULED ALERTS

ALERTING VIEWS

In the Alerts section of the NotesTracker Repository database, there are some views for viewing and managing e-mail alerting, for all of the databases that are being logged to this repository. These views are circled in red in the following illustration:

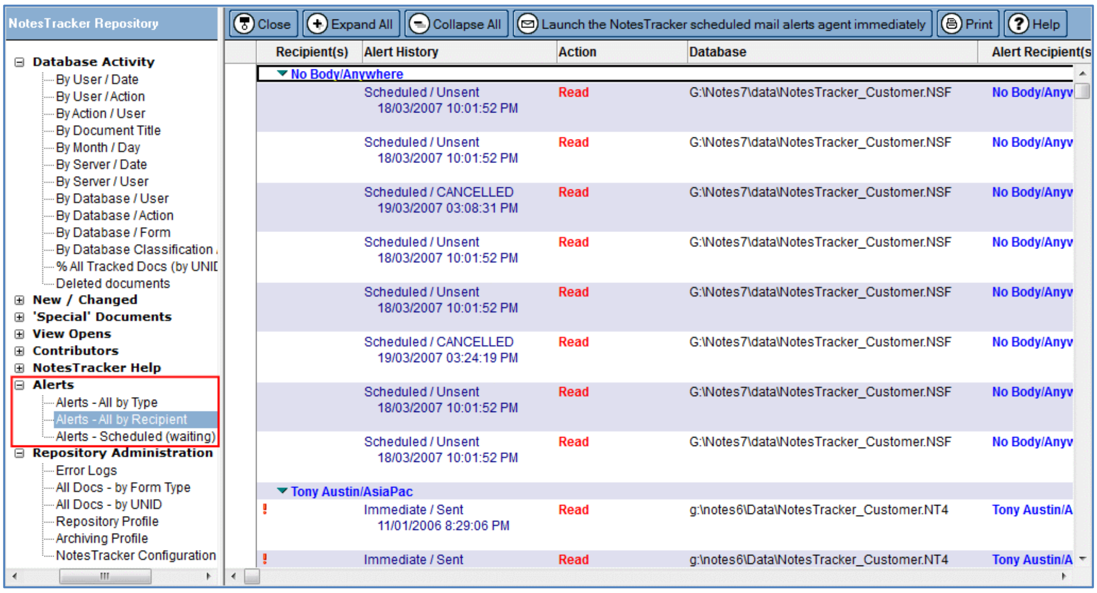


Type	Alert History	Action	Database	Alert Recipient(s)
1 - Immediate / Sent				
!	Immediate / Sent 11/01/2006 8:29:06 PM	Read	g:\notes6\Data\NotesTracker_Customer.NT4	Tony Austin/AsiaPac
!	Immediate / Sent 11/01/2006 7:33:49 PM	Read	g:\notes6\Data\NotesTracker_Customer.NT4	Tony Austin/AsiaPac
	Immediate / Sent 11/01/2006 8:24:39 PM	Read	g:\notes6\Data\NotesTracker_Customer.NT4	Tony Austin/AsiaPac
	Immediate / Sent 11/01/2006 8:27:02 PM	Read	g:\notes6\Data\NotesTracker_Customer.NT4	Tony Austin/AsiaPac
3 - Scheduled / Unsent				
	Scheduled / Unsent 18/03/2007 10:01:52 PM	Read	G:\Notes7\data\NotesTracker_Customer.NSF	No Body/Anywhere
	Scheduled / Unsent 18/03/2007 10:01:52 PM	Read	G:\Notes7\data\NotesTracker_Customer.NSF	No Body/Anywhere
	Scheduled / Unsent 18/03/2007 10:01:52 PM	Read	G:\Notes7\data\NotesTracker_Customer.NSF	No Body/Anywhere
	Scheduled / Unsent 18/03/2007 10:01:52 PM	Read	G:\Notes7\data\NotesTracker_Customer.NSF	No Body/Anywhere
	Scheduled / Unsent 18/03/2007 10:01:52 PM	Read	G:\Notes7\data\NotesTracker_Customer.NSF	No Body/Anywhere
	Scheduled / Unsent	Read	G:\Notes7\data\NotesTracker_Customer.NSF	No Body/Anywhere

It is important to note that the rows in these views represent normal Usage Log documents, not some sort of special alerting documents. Do not make the mistake of deleting these documents, thinking that you are just deleting information about alerts. If you delete any of them, you are losing valuable usage tracking history.

The view shown above ("**Alerts – All by Type**") gives you an overall picture of the alerting progress, categorized by alert status: whether "immediate" or "Scheduled" alerts, and their fate (successfully sent, yet unsent, in error, or cancelled).

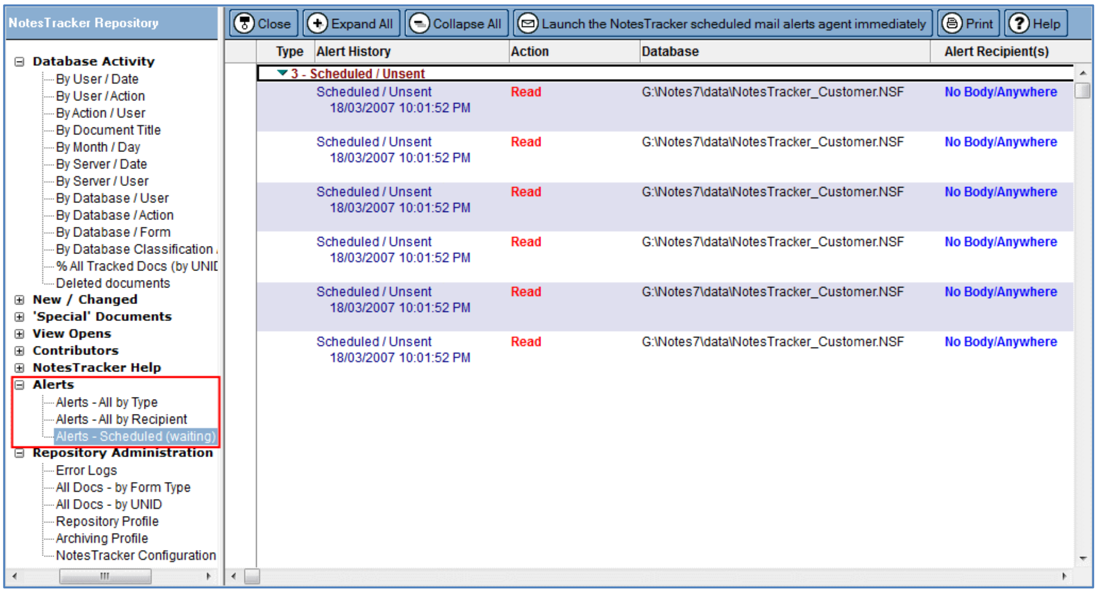
The second view (“**Alerts – All by Recipient**”) also displays every alert, but categorized by recipient:



Recipient(s)	Alert History	Action	Database	Alert Recipient(s
No Body/Anywhere				
	Scheduled / Unsent 18/03/2007 10:01:52 PM	Read	G:\Notes7\data\NotesTracker_Customer.NSF	No Body/Anyv
	Scheduled / Unsent 18/03/2007 10:01:52 PM	Read	G:\Notes7\data\NotesTracker_Customer.NSF	No Body/Anyv
	Scheduled / CANCELLED 19/03/2007 03:08:31 PM	Read	G:\Notes7\data\NotesTracker_Customer.NSF	No Body/Anyv
	Scheduled / Unsent 18/03/2007 10:01:52 PM	Read	G:\Notes7\data\NotesTracker_Customer.NSF	No Body/Anyv
	Scheduled / Unsent 18/03/2007 10:01:52 PM	Read	G:\Notes7\data\NotesTracker_Customer.NSF	No Body/Anyv
	Scheduled / CANCELLED 19/03/2007 03:24:19 PM	Read	G:\Notes7\data\NotesTracker_Customer.NSF	No Body/Anyv
	Scheduled / Unsent 18/03/2007 10:01:52 PM	Read	G:\Notes7\data\NotesTracker_Customer.NSF	No Body/Anyv
	Scheduled / Unsent 18/03/2007 10:01:52 PM	Read	G:\Notes7\data\NotesTracker_Customer.NSF	No Body/Anyv
Tony Austin/AsiaPac				
!	Immediate / Sent 11/01/2006 8:29:06 PM	Read	g:\notes6\Data\NotesTracker_Customer.NT4	Tony Austin/A
!	Immediate / Sent	Read	g:\notes6\Data\NotesTracker_Customer.NT4	Tony Austin/A

We will see in a moment how this view can help with the management of scheduled alerts.

The third view (“**Alerts – Scheduled (waiting)**”) displays the Usage Log documents for only those actions which have been scheduled to send alert e-mails and which have not yet been sent. That ism they are awaiting the next scheduled running of the agent (the name of which is “**Send NotesTracker scheduled mail alerts**”). In other words, it gives you a snapshot of the alerts that should be sent out next. It will look something like this:



Type	Alert History	Action	Database	Alert Recipient(s)
3 - Scheduled / Unsent				
	Scheduled / Unsent 18/03/2007 10:01:52 PM	Read	G:\Notes7\data\NotesTracker_Customer.NSF	No Body/Anywhere
	Scheduled / Unsent 18/03/2007 10:01:52 PM	Read	G:\Notes7\data\NotesTracker_Customer.NSF	No Body/Anywhere
	Scheduled / Unsent 18/03/2007 10:01:52 PM	Read	G:\Notes7\data\NotesTracker_Customer.NSF	No Body/Anywhere
	Scheduled / Unsent 18/03/2007 10:01:52 PM	Read	G:\Notes7\data\NotesTracker_Customer.NSF	No Body/Anywhere
	Scheduled / Unsent 18/03/2007 10:01:52 PM	Read	G:\Notes7\data\NotesTracker_Customer.NSF	No Body/Anywhere
	Scheduled / Unsent 18/03/2007 10:01:52 PM	Read	G:\Notes7\data\NotesTracker_Customer.NSF	No Body/Anywhere

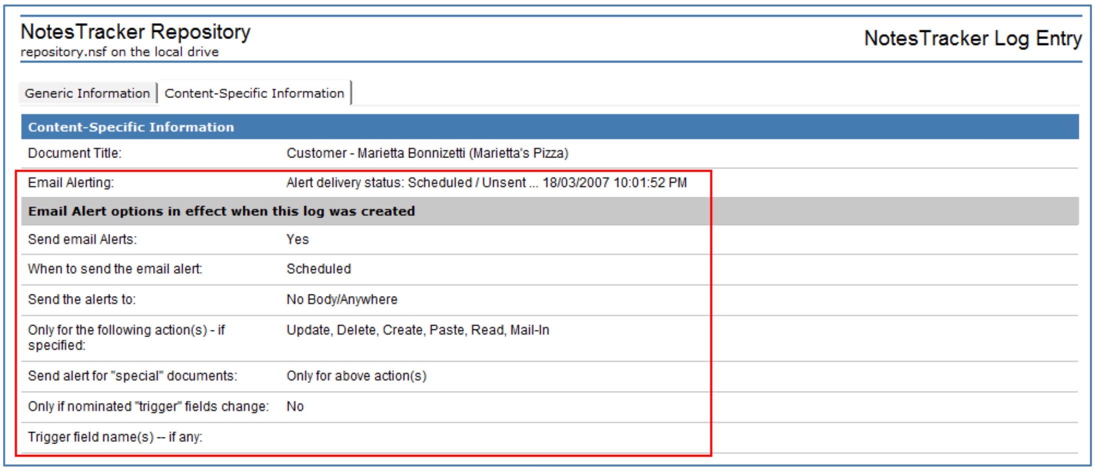
If all goes well, all of these entries should disappear – from the view, not from the Repository database – at the next scheduling agent run.

MANAGING FAILURES IN NOTESTRACKER E-MAIL ALERTING

What can go wrong with e-mail alerting, and how can problems be managed?

In the case of **“Immediate” alerting**, as described earlier there will be (in most cases) the “no match found” dialog box presented to the user, caused by incorrect recipient addresses being entered into the tracked database’s NotesTracker Configuration document. This should not happen (if the administrator does adequate testing to ensure that valid mail addresses are entered). Even if this does occur, the error is obvious as an “in your face” dialog box that should lead to an early amendment of the Configuration Document.

With **“Scheduled” alerting**, the resolution is not quite so obvious. Firstly, you will start getting agent error messages on the Domino console similar to the one shown a page or two earlier. You should be able to correlate that with one of the rows in the third view, let’s say the first one (for 18/03/2007 10:01:52 PM), and open the Usage Log document to see information in the “E-mail Alerting” section of the log, circled in red in the following illustration:



NotesTracker Repository
repository.nsf on the local drive

NotesTracker Log Entry

Generic Information | Content-Specific Information

Content-Specific Information

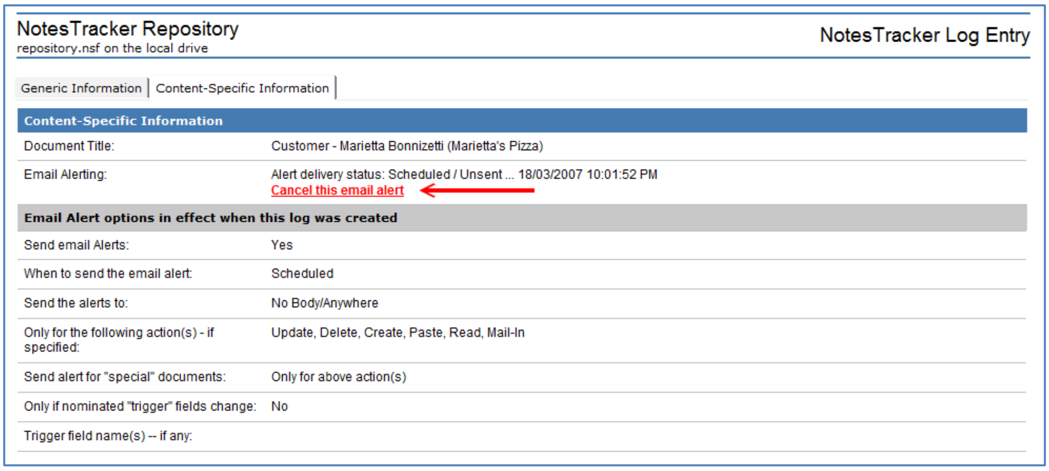
Document Title:	Customer - Marietta Bonnizetti (Marietta's Pizza)
Email Alerting:	Alert delivery status: Scheduled / Unsent ... 18/03/2007 10:01:52 PM
Email Alert options in effect when this log was created	
Send email Alerts:	Yes
When to send the email alert:	Scheduled
Send the alerts to:	No Body/Anywhere
Only for the following action(s) - if specified:	Update, Delete, Create, Paste, Read, Mail-In
Send alert for "special" documents:	Only for above action(s)
Only if nominated "trigger" fields change:	No
Trigger field name(s) -- if any:	

Let’s suppose that there is a problem in sending this particular alert (the Send command in the agent fails, probably due to an incorrect recipient address). This effect of this is that this row remains in the view, longer that a single agent schedule interval and potentially indefinitely. This will result in a console error message being generated it once at each agent run interval.

Assuming that sooner or later you realize that this row will never generate an alert e-mail, you will want to have the row removed from the view. But you must not simply delete the row (that is, delete the Usage Log document) as explained previously!

The solution is to follow the following simple procedure:

1. Open the Usage Log document, and click on the ‘Content-Specific Information’ tab:



NotesTracker Repository
repository.nsf on the local drive

NotesTracker Log Entry

Generic Information | Content-Specific Information

Content-Specific Information	
Document Title:	Customer - Marietta Bonnizetti (Marietta's Pizza)
Email Alerting:	Alert delivery status: Scheduled / Unsent ... 18/03/2007 10:01:52 PM Cancel this email alert
Email Alert options in effect when this log was created	
Send email Alerts:	Yes
When to send the email alert:	Scheduled
Send the alerts to:	No Body/Anywhere
Only for the following action(s) - if specified:	Update, Delete, Create, Paste, Read, Mail-In
Send alert for "special" documents:	Only for above action(s)
Only if nominated "trigger" fields change:	No
Trigger field name(s) -- if any:	

The Email Alerting field shows the alert status of this Usage Log entry is “**Scheduled / Unsent**”

2. If you have the [Admin] role in the NotesTracker Repository’s ACL, you will be able to see the line containing a clickable link labeled “**Cancel this email alert**”

3. Click on this link. The alert delivery status should change to “**Scheduled / Cancelled**“ appended with the date and time that you clicked the link.

4. The ‘Cancel this email alert’ link will not be visible next time the Usage Log document is opened. This prevents repeated cancellations of any given alert.

5. It would be nice – if you can work out who is/are the proper alert recipient(s) – to contact the alert recipient(s) and explain that you have had to cancel the alert.

6. It is suggested that, as a courtesy and if you are able to establish the correct e-mail addresses, you send an e-mail message containing a Doclink to the Usage Log document so that the recipient(s) can examine the nature of the alert.

7. See to it that the relevant NotesTracker Configuration document is amended so that the error does not continue.

Using this rather painless approach, maybe with modifications to cater for variations in the error cause, you should be able to resolve alerting errors fairly easily.

MANAGING SCHEDULED NOTESTRACKER ALERTS IN DATABASES WITH HIDDEN DESIGNS

Sometimes Notes/Domino application databases are used which have hidden designs. Two such are **SDMS** and **CAPTURE –** applications freely available from the AGECOM website:

SDS: https://www.agecom.com.au/sds

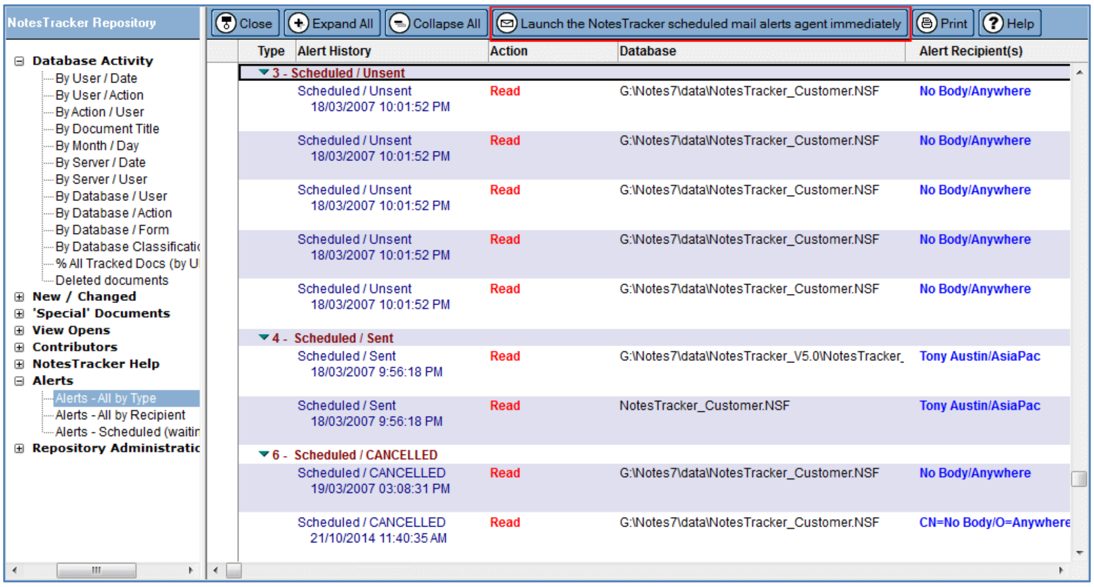
Capture: https://www.agecom.com.au/capture

You should download these and try them out. They both incorporate NotesTracker. As well as possibly finding them useful in their own right, they will be used as good examples of the matter we are now going to examine.

Since their designs are hidden, and the NotesTracker email alerting agent in each of them is turned off (unscheduled) by default, you cannot access their settings directly via the Lotus Domino Designer client.

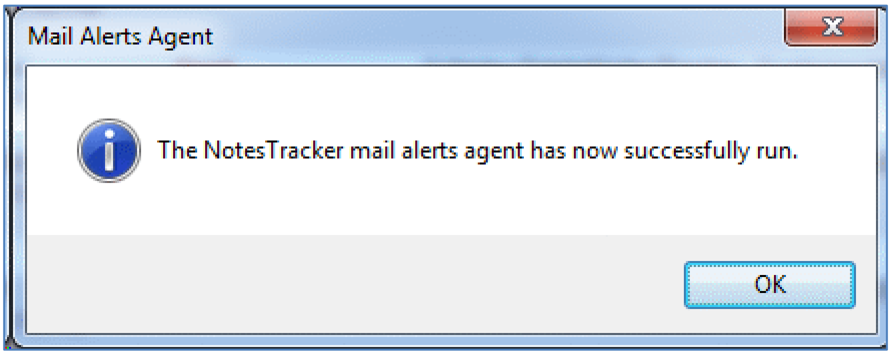
The ability to manage the settings for these agents and to enable / disable them is available via the ‘Scheduled Agents’ section of the NotesTracker Configuration Settings form. Please refer to the ‘Managing Scheduled Agents’ section further down this guide for more information.

A button has also been provided in the Alerts views to run the scheduled alerts agent immediately The button is labeled “**Launch the NotesTracker scheduled mail alerts agent immediately**” and is circled in red in the following illustration:



Type	Alert History	Action	Database	Alert Recipient(s)
3 - Scheduled / Unsent				
	Scheduled / Unsent 18/03/2007 10:01:52 PM	Read	G:\Notes7\data\NotesTracker_Customer.NSF	No Body/Anywhere
	Scheduled / Unsent 18/03/2007 10:01:52 PM	Read	G:\Notes7\data\NotesTracker_Customer.NSF	No Body/Anywhere
	Scheduled / Unsent 18/03/2007 10:01:52 PM	Read	G:\Notes7\data\NotesTracker_Customer.NSF	No Body/Anywhere
	Scheduled / Unsent 18/03/2007 10:01:52 PM	Read	G:\Notes7\data\NotesTracker_Customer.NSF	No Body/Anywhere
	Scheduled / Unsent 18/03/2007 10:01:52 PM	Read	G:\Notes7\data\NotesTracker_Customer.NSF	No Body/Anywhere
4 - Scheduled / Sent				
	Scheduled / Sent 18/03/2007 9:56:18 PM	Read	G:\Notes7\data\NotesTracker_V5.0\NotesTracker_	Tony Austin/AsiaPac
	Scheduled / Sent 18/03/2007 9:56:18 PM	Read	NotesTracker_Customer.NSF	Tony Austin/AsiaPac
6 - Scheduled / CANCELLED				
	Scheduled / CANCELLED 19/03/2007 03:08:31 PM	Read	G:\Notes7\data\NotesTracker_Customer.NSF	No Body/Anywhere
	Scheduled / CANCELLED 21/10/2014 11:40:35 AM	Read	G:\Notes7\data\NotesTracker_Customer.NSF	CN=No Body/O=Anywhere

When you click this button, the otherwise “dead” mail messages are processed immediately by the mail alert agent, and you should see the “Success” dialog box, like this:



The NotesTracker mail alerts agent has now successfully run.

After this view is refreshed, you should see that the otherwise “dead” alert messages have been dispatched:

This button is provided (in the form of a shared action) in the NotesTracker Repository database, so your developer might be able to arrange for this capability to be similarly incorporated in databases that you use which have their designs hidden.

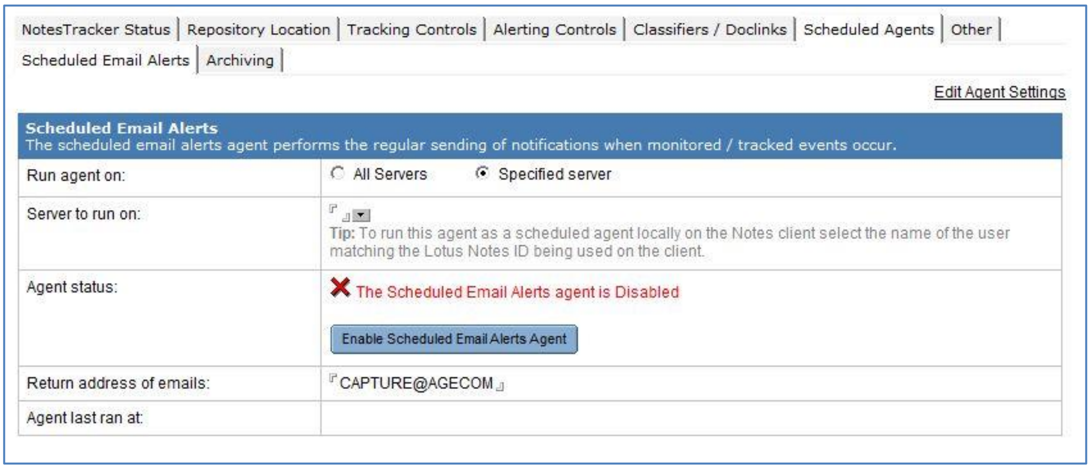
MANAGING SCHEDULED NOTESTRACKER AGENTS

Scheduled Lotus Notes / Domino agents are normally managed using the Lotus Domino Designer client. However in some applications, where the design is hidden, access to the agent settings is not available. For example the NotesTracker evaluation application and some NotesTracker related sample applications all have their design hidden.

NotesTracker comes with scheduled agents that perform the following functions:

- Sending of email alerts
- Archiving of Usage Log documents

By default these agents are turned off (unscheduled). The ability to manage the settings for these agents and to enable / disable them is available via the ‘Scheduled Agents’ section of the NotesTracker Configuration Settings form.

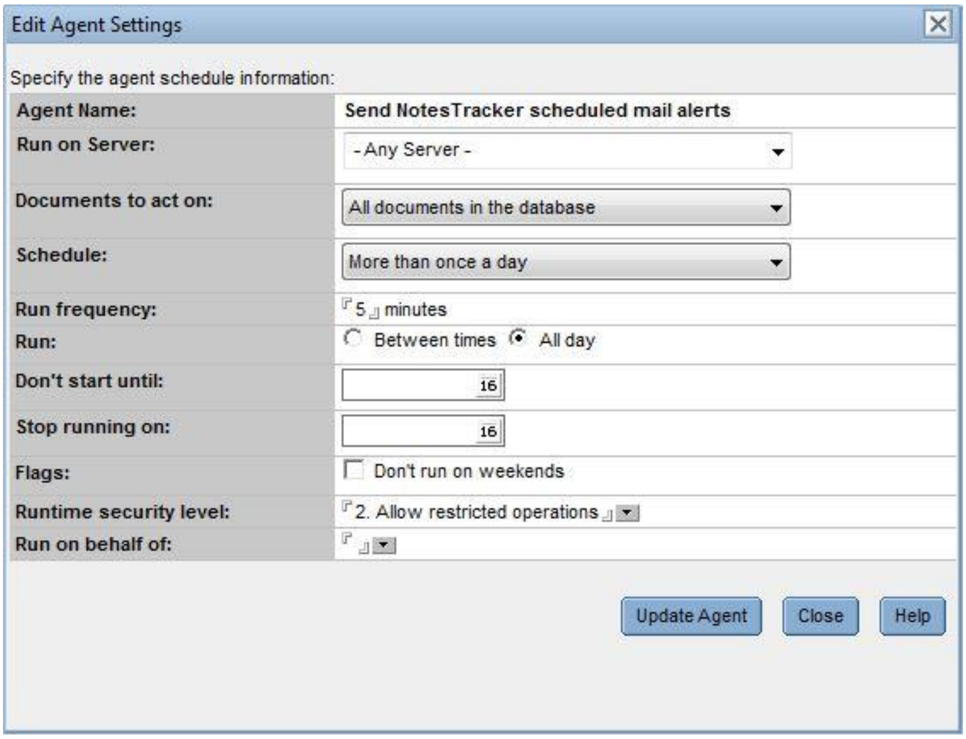


Scheduled Email Alerts
The scheduled email alerts agent performs the regular sending of notifications when monitored / tracked events occur.

Run agent on:	○ All Servers ● Specified server
Server to run on:	Tip: To run this agent as a scheduled agent locally on the Notes client select the name of the user matching the Lotus Notes ID being used on the client.
Agent status:	✗ The Scheduled Email Alerts agent is Disabled
Return address of emails:	CAPTURE@AGECOM
Agent last ran at:	

Under the ‘Scheduled Agents’ tab you will find separate tabs for each available scheduled agent.

To edit the settings for a scheduled agent (such as when to run the agent, how often, which server, etc) click the ‘Edit Agent Settings’ option near the top right-hand corner of the section. The following box will open:



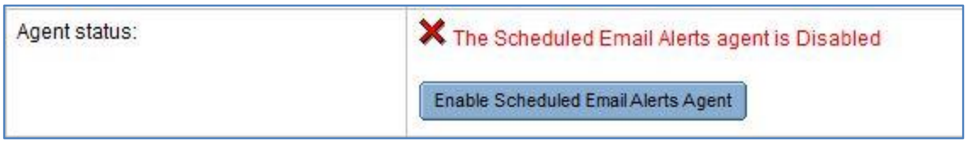
Edit Agent Settings

Specify the agent schedule information:

Agent Name:	Send NotesTracker scheduled mail alerts
Run on Server:	- Any Server -
Documents to act on:	All documents in the database
Schedule:	More than once a day
Run frequency:	5 minutes
Run:	○ Between times ● All day
Don't start until:	16
Stop running on:	16
Flags:	□ Don't run on weekends
Runtime security level:	2. Allow restricted operations
Run on behalf of:	

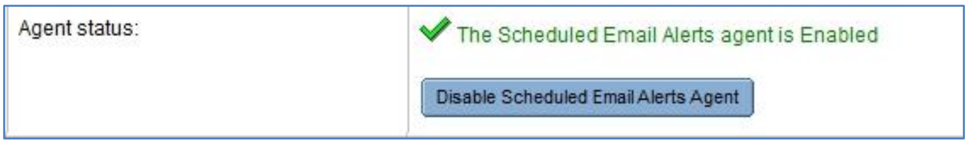
Once you’ve finished configuring the settings for the agent click the ‘Update Agent’ button at the bottom of the box.

If the scheduled agent is currently disabled the status will be shown with a button to enable the agent:



Agent status:	✗ The Scheduled Email Alerts agent is Disabled

If the scheduled agent is currently enabled the status will be shown with a button to disable the agent:



Agent status:	✓ The Scheduled Email Alerts agent is Enabled

ERROR LOGGING

If an error occurs during NotesTracker processing an error log document is written to the NotesTracker Repository database.

These error log documents can be viewed in the ‘Error Logs’ view of the Repository database. The error logs are sorted and categorized by the database the error occurred in, and then further sorted by the date / time the error occurred in date descending order.

It is recommended this view is periodically opened to check for any errors that are occurring and may need attention.

Error logs can be easily sent to the developer of this product by selecting one or more log documents in the view then click the ‘Send selected logs to AGECOM Support’ button at the top of the view.

If you use Lotus Notes for your email you can send the logs immediately otherwise you can choose the option to copy the information for the selected logs to the Windows clipboard and paste them in an email to AGECOM Support using your preferred email client.

NOTESTRACKER DATABASE SIZE, VIEW INDEXES, AND PERFORMANCE CONSIDERATIONS

THE BUILD-UP OF USAGE LOG DOCUMENTS, AND VIEW INDEX OVERHEADS

As a very rough approximation, the database size increases at 1.5KB to 2KB per log document. The growth rate needs to be monitored, and you should devise an appropriate archive-and-purge strategy if disk space is a worry. How frequently you purge log documents should primarily be determined by the length of time – typically a number of months (or even years) -- for which you wish to retain usage metrics.

Of course, it’s not only document contents that take up space in a database. Keep in mind that view indexes will have a major impact on database growth, rather than the relatively small amount of data stored in the log documents. To reduce Notes Client view opening overheads (and Domino server workload needed to maintain the view indexes), the number of sorted view columns has been kept reasonably low. However, you may wish to alter the view designs to decrease the number of sorted view columns even further, or to make other changes that balance view opening times against indexing overheads to your satisfaction.

As a guide, one user of NotesTracker found that some 60,000 Usage Log entries used close to 1 GB of disk space, and you may get similar results.

DISK SPACE MANAGEMENT – ARCHIVING AGENT

An archiving agent is available and can be run on an as-required or scheduled basis, giving you the control you need over database size. The archive agent is discussed a little further on.

WEB BROWSER USAGE TRACKING PERFORMANCE OVERHEADS

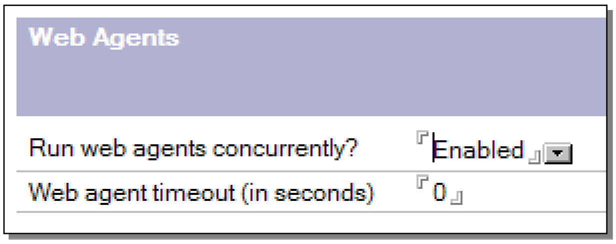
Web browser usage tracking brings some new performance matters to consider. With a web application – for Domino or any other web server – there is an extra processing burden placed on the web server by *each and every browser interaction* with the server. This applies to HTTP GET/POST operations, running of CGI programs on the server, running or Java servlets, and so on. You need to be cognizant of the fact that whenever you have NotesTracker web tracking active for a given database, this will add somewhat to the Domino server load, as will the additional NotesTracker Usage Log entries (which are ordinary Notes documents in the NotesTracker repository database). Such things should be kept in mind and become part of your Domino capacity planning and performance monitoring operations.

Some other related performance considerations are discussed several pages above, in the section on Web Browser usage tracking.

TIP – ALLOWING MORE WEB AGENTS TO RUN CONCURRENTLY

Unless you change the default setting, Domino web agents run serially (consecutively, one at a time), which can cause requests to queue up and might lead to poor response times.

You can experiment with this by editing the **Internet Protocols** section of the Server Document:



Web Agents

Run web agents concurrently?	Enabled
Web agent timeout (in seconds)	0

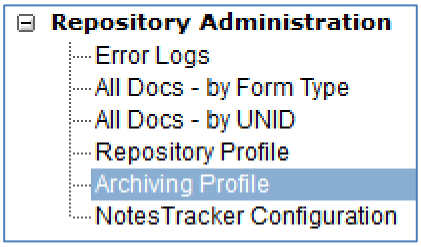
Allowing web agents to run concurrently might improve the user's perception of Domino responsiveness, but equally might lead to server workload problems. Monitor the effect (on both user response times and server workload) for a suitable period, and turn this feature back off if the net results are not satisfactory.

MANAGING THE USAGE LOG – THE NOTESTRACKER ARCHIVING AGENT

An important feature of NotesTracker is the NotesTracker Periodic Archive agent.

SETTING UP NOTESTRACKER ARCHIVING

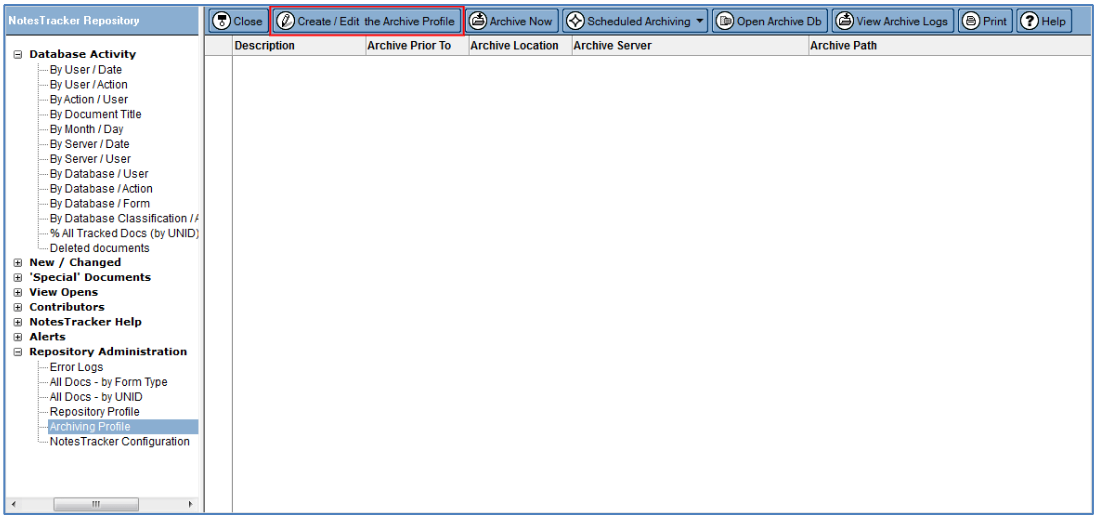
To setup NotesTracker archiving click the ‘Archiving Profile’ on the left-hand navigation menu of the NotesTracker Repository database.



The archiving view is designed to hold a single NotesTracker Archive Profile document. You must create and configure this profile document before the agent will run.

Note: you must have **Manager** or **Designer** access level rights to be able to view the action buttons described below for setting up and running the archiving agent.

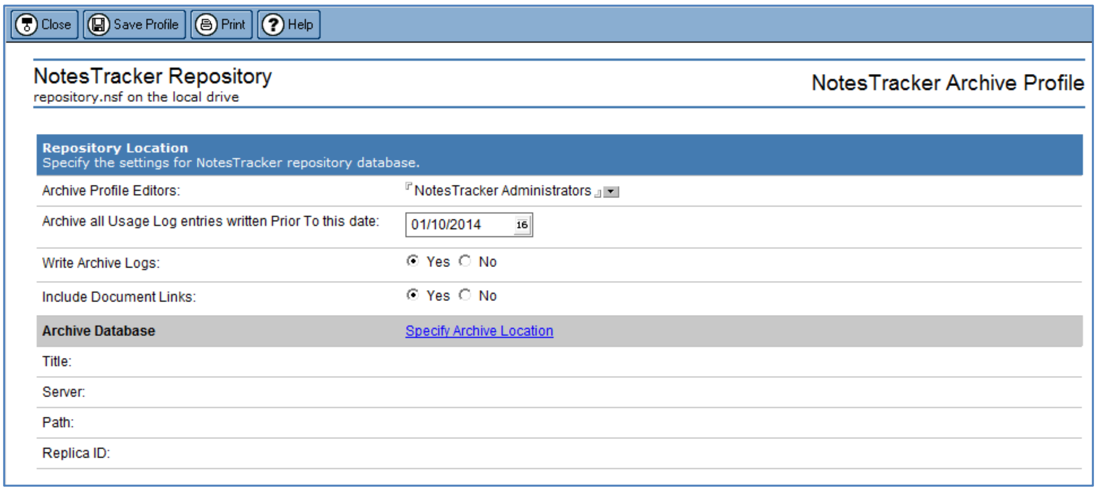
The very first time that you click on the Archiving Control button, the archiving view will be empty, like this:



Description	Archive Prior To	Archive Location	Archive Server	Archive Path

Click on the “**Create / Edit the Archive Profile**” action button (circled in red) to set up the profile document.

While it is being edited, the **NotesTracker Archive Profile document** will look like this:



NotesTracker Repository
repository.nsf on the local drive

NotesTracker Archive Profile

Repository Location
Specify the settings for NotesTracker repository database.

Archive Profile Editors:	NotesTracker Administrators
Archive all Usage Log entries written Prior To this date:	01/10/2014
Write Archive Logs:	● Yes ○ No
Include Document Links:	● Yes ○ No
Archive Database	Specify Archive Location
Title:	
Server:	
Path:	
Replica ID:	

You need to specify the following information:

1. The person(s) who are allowed to edit the NotesTracker Archive Profile document.

2. The date **prior to which** any Usage Log documents in the NotesTracker database are to be archived.

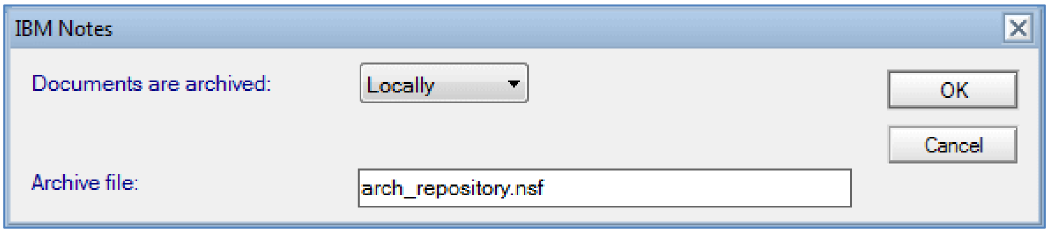
   The default is the first day of the current month. It is suggested that your archive be made on a monthly basis, but this is entirely up to you (and is largely dependent on the rate at which Usage Log documents are generated, plus how long you wish to keep them online for the viewing of historical usage patterns and trends).

3. If logs should be written for each archiving session.

4. If links to archived documents should be included in archiving logs.

5. The **location** of the archive database. This is made up of a **server** component and a **file** component.

   You click on the **Specify Archive Location** link to provide the names of these components, and should see the following dialog box:



IBM Notes

Documents are archived: Locally

Archive file: arch_repository.nsf

Documents can be archived either “Locally” or on a Domino server.

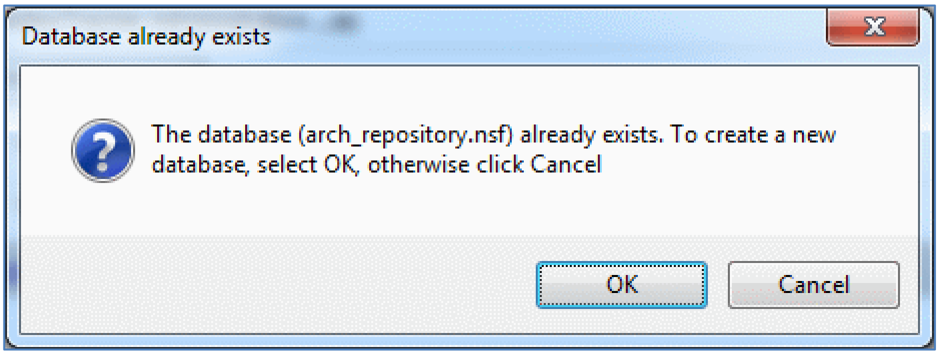
The archive file name is the path to the Notes database which will receive the archived Usage Log documents. The file name can be preceded, in the normal fashion, by a directory path (in the proper fashion for the operating system, and relative to the Domino data directory).

Examples are: **arch_repository.nsf** or **archive\2014\ arch_NotesTracker_June.nsf**

> **Note**: the archive database can be any database to which you have rights to add documents. However, it will make life easier for you if the archive database is a copy of the NotesTracker Database (which is the default), since this will contain all the regular NotesTracker views. This enables you to view the archived documents without having to add suitably designed views to the archive database.

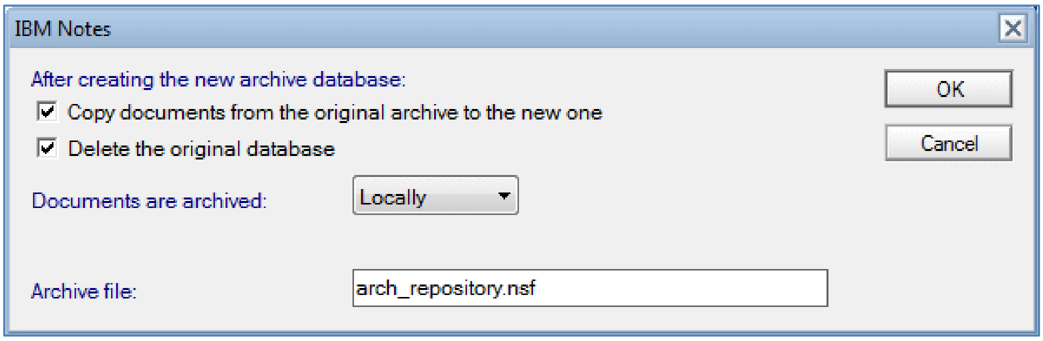
When you click the OK button, the archive database is created if it does not already exist. (This takes a few seconds.)

If you click the link after the archive database has been created, you will see the following message:



The database (arch_repository.nsf) already exists. To create a new database, select OK, otherwise click Cancel

Click cancel to abort the creation of a new archive database, otherwise click Ok to continue. If you click the Ok button the following dialog will then be displayed:



After creating the new archive database:

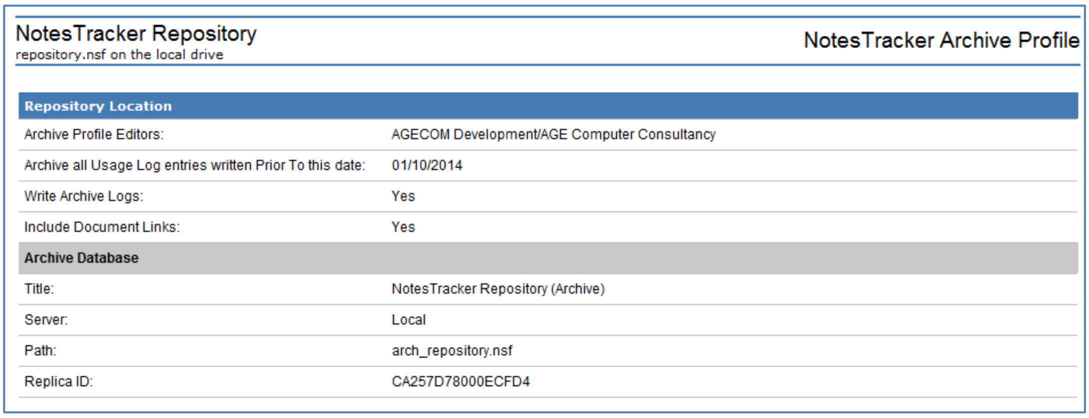
☑ Copy documents from the original archive to the new one

☑ Delete the original database

Documents are archived: Locally

Archive file: arch_repository.nsf

A completed Archive Profile looks something like this:

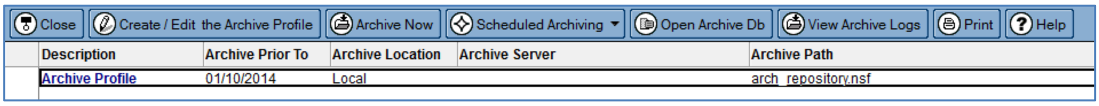


NotesTracker Repository
repository.nsf on the local drive

NotesTracker Archive Profile

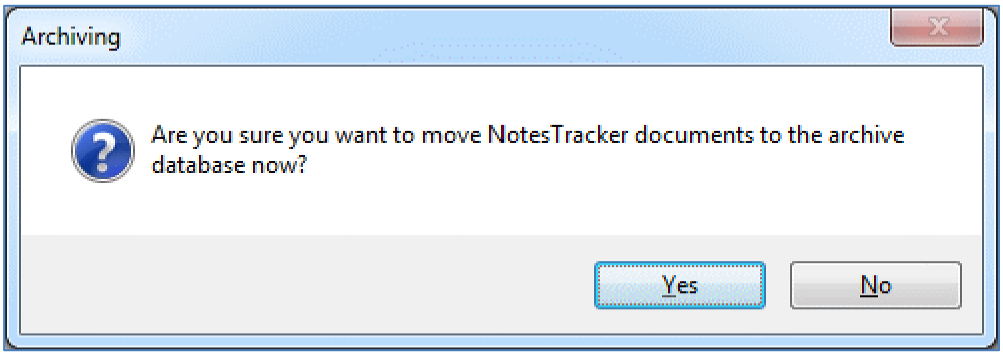
Repository Location	
Archive Profile Editors:	AGECOM Development/AGE Computer Consultancy
Archive all Usage Log entries written Prior To this date:	01/10/2014
Write Archive Logs:	Yes
Include Document Links:	Yes
Archive Database	
Title:	NotesTracker Repository (Archive)
Server:	Local
Path:	arch_repository.nsf
Replica ID:	CA257D78000ECFD4

When you return to the NotesTracker Archiving view, you should see something like this:



Description	Archive Prior To	Archive Location	Archive Server	Archive Path
Archive Profile	01/10/2014	Local		arch_repository.nsf

At this stage, you can click the **"Archive Now"** action bar button to cause the archiving agent to run immediately. You are asked for confirmation:

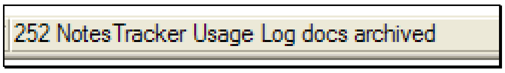


Archiving

Are you sure you want to move NotesTracker documents to the archive database now?

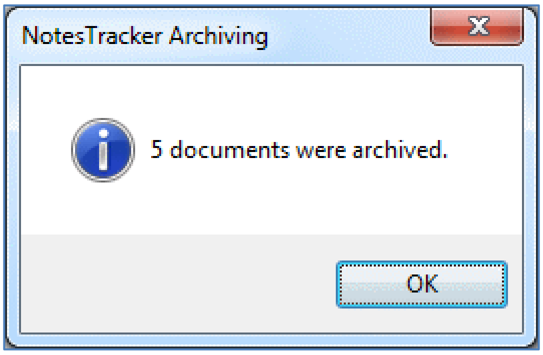
If you confirm by clicking the "Yes" button, then the agent runs on your Notes Client workstation.

Watch the status line to follow the agent's progress. This might only take a few seconds or a few minutes (depending, of course, on how many documents are to be archived), but you will see something like:



252 NotesTracker Usage Log docs archived

Upon successful completion you will see a dialog box like the following:



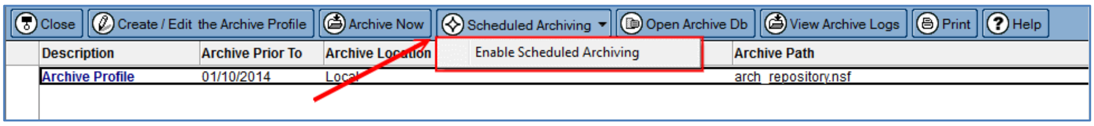
NotesTracker Archiving

5 documents were archived.

Alternatively, you can **schedule the Archiving Agent to run later**. You do so using normal techniques for setting up the agent's schedule (daily, weekly, monthly, etc) and for activating the agent... It is beyond the scope of this NotesTracker Guide to explain this any further.

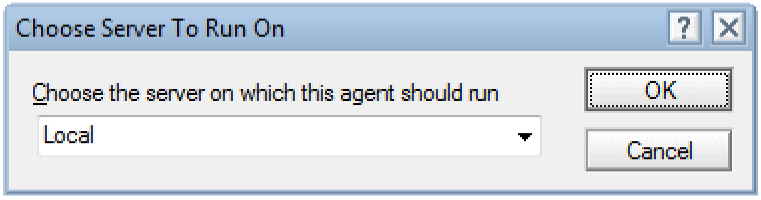
If you don't run the agent immediately (via the "Archive Now" button), then you can schedule it to run automatically at some designated periodic interval or date/time.

To enable the agent click on the 'Scheduled Archiving' button and select 'Enable Scheduled Archiving' from the dropdown menu. You should set the schedule period and date/time in the normal fashion, as described in the Lotus documentation.



Description	Archive Prior To	Archive Location	Archive Path
Archive Profile	01/10/2014	Local	arch_repository.nsf

If this is the first time the archiving agent is being enabled the following prompt will appear:



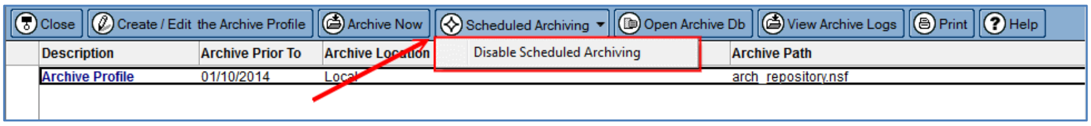
Choose Server To Run On

Choose the server on which this agent should run

Local

Select or enter the name of the Domino server the scheduled archiving should be run on.

To disable the agent click on the 'Scheduled Archiving' button and select 'Disable Scheduled Archiving' from the dropdown menu.



Description	Archive Prior To	Archive Location	Archive Path
Archive Profile	01/10/2014	Local	arch_repository.nsf

FACTORS DETERMINING THE FREQUENCY OF ARCHIVING

How often you run the Archiving Agent is up to you, and will depend on such factors as:

- How long you wish to keep NotesTracker Usage Log documents in the repository database in order to analyze usage patterns and trends, etc;
- What restrictions you have on disk storage space on your Domino server(s); and
- How rapidly Usage Log entries are being added to the repository database.
- Such factors will vary from organization to organization; application to application, and from time to time (there will be peaks and troughs in usage). This means that the frequency of archiving for any given usage tracking environment might be days, weeks or months.

  As mentioned earlier, on NotesTracker user reported that some 60,000 Usage Log documents took up 1 GB of disk space.

VIEWING THE ARCHIVED USAGE LOG DOCUMENTS

To view archived Usage Log documents, as a convenience you can click on the “Open current Archive Db” button to open the archive database that is specified in the Archive Profile document.

Click on the “Open current Archive Db” button to open the database specified in the Archive Profile document.

CUSTOMIZING WHEN THE SCHEDULED ARCHIVING AGENT RUNS

You can customize when the archiving agent runs and how often via the NotesTracker Configuration Settings form. Please refer to the ‘Managing Scheduled NotesTracker Agents’ topic earlier in this guide for more information.