Essential Tools Return on Investment (ROI) Statement

Essential Tools (ET) is an award-winning robust server-based Domino Administration tool designed to assist Domino administrators perform their tasks significantly more efficiently and also allow them to fulfill certain tasks that the infrastructure did not allow them to do so before. ET excels in Database, ACL and User management functionalities.

The robustness of ET stems from the basic architectural fact that the Domino Infrastructure information (Database, ACL, etc.) is collected in a centralized single point (et.nsf) and API-based ET server tasks execute enterprise-wide transactions at high speeds all in the background.

An additional robustness is achieved through replicating the transaction documents in the background, thus gaining the same access to remote servers with slow network connections as if they were in the same high-speed local are network (LAN).

Finally, the transaction creation and execution is fully secured with well-defined rules and all transaction processing is fully logged for audit trail purposes.

Based on the above architectural features and consequent behavior, ET has a high ROI which can be expressed in several different functional ways. ACL Enforcement, Termination and Database Design Management are the functions quantified in the following sections. Additional, the value of the ET feature of "Copy Entry" is explored, as well the enhancement of reliability that ET lends to the AdminP functionality of the Domino administration is addressed.

1. **ACL Enforcement**

With the introduction of ND6, some may argue that no third-party tools are needed anymore for ACL Management since each server can have full administrator(s) with full ACL rights to all the databases on that server.

While the Full Administrator feature removes the issue of not having full ACL access to databases, it creates a number of new problems.

a. If you assign a senior administrator as the full administrator, then the senior administrator can be bogged down with mundane & manual tasks such as changing database ACLs all the time – tasks that can easily be filled by less senior personnel.
b. If you assign a junior administrator as the full administrator, then the junior administrator will have much more privilege than he/she normally needs and potential for making mistakes will increase.
c. In either case, the full auditing of who-has-done-what will be cumbersome, if even possible since there is no comprehensive, native auditing capability in Domino.

ET provides a fully privilege-based and fully logged ACL management facility that along makes it a favorite among Domino administrators.

In order to quantify saved time resulting from ET ACL Management, let's have a look at the following hypothetical task:
If a Domino administrator were to add one ACL entry to 20 databases on 100 Domino servers in the network, how many man-hours would be needed?

Case A:
We are assuming that network speed is reasonably fast and uniform for all servers. Therefore, manually adding one ACL entry to one database involves the following steps.

a. Open database (6 seconds)
b. Open ACL (3 seconds)
c. Add new ACL name (4 seconds)
d. Specify access level (2 seconds)
e. Specify user type (2 seconds)
f. Review/Verify you have entered the data correctly (3 seconds)
g. Click OK and wait for it to take effect (2 seconds)

Performing this very basic ACL operation takes 22 seconds per database – assuming that the administrator makes no mistakes along the way. The total time adds up to 44,000 seconds, which is 12 hours and 13 minutes during which the administrator must pay undivided attention and is not allowed to make any mistakes. *This equals to about 1 and a half days work of one person.*

Case B:

We are assuming that network speed is reasonably fast and uniform for half of the servers. The other half of the servers (due to slower network speeds) adds another extra 10 seconds per database to complete one operation. Thus, manually adding one ACL entry to one database involves the following steps in the case of the slow servers.

a. Open database (10 seconds)
b. Open ACL (4 seconds)
c. Add new ACL name (4 seconds)
d. Specify access level (2 seconds)
e. Specify user type (2 seconds)
f. Review/Verify you have entered the data correctly (3 seconds)
g. Click OK and wait for it to take effect (7 seconds)

Performing this very basic ACL operation takes 32 seconds per database – assuming that the administrator makes no mistakes along the way.
The databases on the speedy half of the servers will take 22,000 seconds total.
The databases on the slow servers will add up to 32 * 1000 = 32 000 seconds total.
The overall total comes to 22,000 + 32,000 = 54,000 seconds that is 15 hours during which the administrator must pay undivided attention and is not allowed to make any mistakes. *This equals to about 2 full days work of one person*.

In realistic terms, the hypothetical task will take much longer to complete because human beings get tired, take breaks, make mistakes and take more time to correct mistakes. In this manner, one and a half days becomes 2 full business days, 2 days becomes 3 full business days and so forth.

In both Case A and Case B, by using ET it will take no more than 1 minute to create a transaction document (i.e., a request) to do the whole job! The request will then be executed by the ET server tasks right away and replicated to/executed on all the other servers based on the replication schedule without taking time from the Domino administrator.

The value of ET is multiplied when the ACL task defined above is implemented as an ACL Enforcer where the execution is repeated potentially a dozen times per day, whereas the time it takes from the Administrator remains the same – about one minute!

In order to calculate your own savings, all you have to do is to plug in your own database and server numbers to the equation. Also, remember to factor in the number of ACL entries you wish to change in given task.

2. Termination

For "Termination" we elected to just simply list an approved quote from one of the major users of ET - The AT&T Administrator, who states the follow example:

"Essential Tools is an efficient and effective tool in our organization and continuously saves us time/money. Recently ***by using ET, an administrator was able to remove 7,000 users from our Domino environment in a single day****. We calculated the same task* ***using the Admin Client would have taken*** *that administrator approximately 50 dedicated eight-hour days, or* ***2½ months of full-time labor."***

- Frank Costanzo, AT&T

Of course the same savings can be realized for organizations adding individuals as well as terminating.

3. Database Design Management

Template management in today's Domino environments can be a tedious and problematic situation. With users requiring "Manager" access to their mail database in order to use the Out of Office features, they can use alternate database templates and in some cases can even remove the inheritance options from their database. This results in inconsistent appearance of messages, calendaring issues and also will restrict further product enhancement by preventing the implementation of new design features in mail databases. Unfortunately, IBM Notes/Domino does not provide an efficient method of monitoring or managing template inheritance for databases. There is also the case where an environment is to be upgraded to use new templates for mail.

For the purpose of this document, the latter example will be illustrated. In this case there are two options open to the Domino Administrator.

1.) The manual option to check and allocate the correct template for a mail file is time consuming, as it involves several steps and can only be performed on each database individually. The steps involved (using a Domino Administrator client) are:

a. Open Files tab and select database (3 seconds)
b. Select Database Properties (2 seconds)

c. Select Design tab, check Inherit Design box and replace Template name (5 seconds)

The duration for a single database is approximately 10 seconds and this again assumes that there have been no inadvertent mistakes. For a typical mail server of 1,000 mail files, this results in a time of 2 hours 40 minutes. **For an environment of, say, 100 mail servers this would result in approximately 280 hours spent changing the database properties**. The design changes caused by the template change would be applied as the design task runs (typically on a daily basis in the early morning)

2.) IBM provides a tool called Convert that allows administrators to make mass changes to template inheritance and also apply those design changes immediately. This shortens the time spent handling the upgrade. The problem with using the convert tool is that it is extremely resource intensive and is best run when the server is down. For the same 1,000 user server the outage would be in the region of 2 to 3 hours to allow the tool to complete. The challenges with using convert are not only the service outage issue but also the necessity to have desktop access on the server to run the task locally. **For a 100 server environment, the total outage would be in the same range as the time taken to perform the template changes manually.**

Using Essential Tools, the administrator's time taken to perform this action is reduced dramatically. A single request can be generated for an enterprise regardless of the number of servers and the **Essential Tools tasks are performed typically in less than a half hour**, dependant on local configuration and refresh rates. The design changes would be applied as in the manual configuration example by running the design task overnight.
In this example, the 4+ man weeks required to perform the tasks either manually or using the convert task would be reduced to an afternoon at most.

4. Copy Entry

Copy Entry is a unique feature of ET where the ACL and Address Book imprint (including "nested" ACL's) of a Notes user (User A) can be duplicated for another Notes user (User B) throughout the enterprise via a single ET request.

For example, let's assume that User B has just joined the company and she requires exactly the same access rights as User A to the Domino environment including group memberships, ACL access levels and ACL roles. Copy Entry request basically does the following.

a. Adds User B to every group that User A is a member of (in Domino Directory).
b. Adds User B to every ACL (on every Database on every server) that User A is listed with the same access level and roles as User A.

Another scenario where Copy Entry request is an invaluable help is the case of a name change. If we want the new name to coexist with the old name for a period of time (Ex. married name and maiden name), then applying Copy Entry request will duplicate the imprint of the old name into the new name in the manner described above. The coexistence may be needed for prudent reasons so as not to cause any blackout period caused by possibly incomplete name change process with AdminP (see next section). After the coexistence period is over, the old name can be safely removed (i.e., terminated) without any down time for the user.

Note: Implementing this feature with native Notes and Domino tools is very difficult since the discovery phase of User A's current ACL access rights is not entirely possible. The reason for that is the fact that the native Database and ACL catalog in Domino does not provide information for all Databases on servers – it excludes the Databases with a certain flag set. For the sake of simplicity, we will overlook this significant fact in favor of the Domino Admin client and AdminP.

How long would it take to copy all the ACL access rights of one Notes user (User A) to a new Notes user (User B) in a Domino environment of 100 servers with an average of 400 Databases per server?

The overall task of **manually** performing Copy Entry function involves two steps: Discovery phase and implementation phase.

a. Discovery phase consists of determining what access (plus Roles) User A has to what Databases on what servers.
b. Implementation phase consists of giving User B the same exact ACL access (plus Roles) as User A.

If we assume that User A is explicitly listed in 5% of all Databases on any/every server, then the discovery phase reduces to opening catalog.nsf (available in R9 and later) on each server and checking the appropriate view in this database to compile one list of all occurrences of User A on that server's Database ACLs. The total is 400 * 5 / 100 = 20 occurrences per server. If opening catalog.nsf takes 10 seconds, and if writing down each occurrence takes 5 seconds, then total discovery phase takes (10 + (5 * 20)) * 100) = 11 000 seconds. This equals to 3 hours of one person in a fast network environment.

For the implementation phase, we have to open each of the Databases in the list compiled in the discovery phase and change its ACL accordingly. We are assuming that network speed is reasonably fast and uniform for all servers. Therefore, manually adding User B ACL entry to one database involves the following steps.

a. Open database (6 seconds)

b. Open ACL (3 seconds)
c. Add User B to the ACL name (3 seconds)
d. Specify access level with Roles if necessary (3 seconds)
e. Specify user type (2 seconds)
f. Review/Verify you have entered the data correctly (3 seconds)
g. Click OK and wait for it to take effect (2 seconds)

One single operation takes a total of 22 seconds. The total time to implement on all 100 servers is 22 * 20 * 100 = 44,000 seconds. **This equals to over 12 hours of one person in a fast network environment.**

The overall total time it **takes to discover and implement Copy Entry function manually is 3 + 12 = 15 hours of work for one person**. This corresponds to almost two full days of one person with no room for any mistake.

Using ET, however, the time to solution is limited to filling up a Copy Entry request form, which should not take more than **30 seconds**!

5. Admin Process Assistance

Admin Process (AdminP) is the native administration utility of Domino server. It provides invaluable help in performing the Domino administration tasks. These tasks include, but are not limited to changing names of Notes users, terminating Notes users, and so forth. **However, AdminP does not work on any database that does not have the Administration Server set properly in the ACL – making it untrustworthy to many Administrators. Therefore, setting the Administration Server on every Database on every server is vital to make AdminP work properly! This problem is experienced by many organizations, which are dependent on AdminP as the primary Domino Administration utility.**

ET becomes very handy at this point by providing a report of Databases without a proper Administration Server setting. **ET** also provides a transaction called ACL Change, which **is used to set the Administration Server on all Databases on a given server or server(s) en masse – with only one point-and-click request form!**

Assuming that 20% of databases on a given server have an improper Administration Server setting, and the environment has 400 servers, and every server has an average of 400 Databases present, how long would it take to fix the entire environment so that AdminP can finally start working properly?

Case A:

We are assuming that network speed is reasonably fast and uniform for all servers. The overall task involves detecting and fixing the all ACLs with improper Administration Server setting. Detecting the faulty Database ACLs among a pile of 400 * 400 = 160,000 Databases, takes its own toll too. For every Database that does have a proper Administration Server setting (a total of 160 000 * 80 / 100 = 128,000) the following 11 seconds has to be spent:

a. Open database (6 seconds)
b. Open ACL (3 seconds)
c. Click on Advanced tab (1 second)
d. Verify the Administration setting (1 second)

This totals to an amount of 1,408,000, which is 391 hours! 391 hours roughly equals to 49 business days of one person for merely detecting faulty ACLs!

Remember that we are assuming that network speed is reasonably fast and uniform for all servers. Therefore, manually updating the Administration Server ACL entry in one database involves the following steps.

a. Open database (6 seconds)
b. Open ACL (3 seconds)
c. Click on Advanced tab (1 second)
d. Click on Server option (1 second)
e. Locate/type the correct Administration Server name (4 seconds)
f. Click on Basics tab (1 seconds)
g. Review/Verify you have entered the data correctly (3 seconds)
h. Click OK and wait for it to take effect (2 seconds)

Performing this very basic ACL operation takes 21 seconds per database – assuming that the administrator makes no mistakes along the way. The total average number of Databases with improper ACL settings on a given server is 400 * 20 / 100 = 80. The total number of Databases with improper ACL settings is 80 * 400 = 32,000. Therefore, total time for fixing the faulty ACLs adds up to 21 * 32,000 = 672,000 seconds. This equals to 186 hours which equals to 23 days of one person!

The overall total of detecting and fixing faulty ACLs equals to 49 + 23 = 72 business days of one person!

Case B:

The overall task involves detecting and fixing the all ACLs with improper Administration Server setting. Detecting the faulty Database ACLs among a pile of 400 * 400 = 160 000 Databases, takes its own toll too. We are assuming that network speed is reasonably fast and uniform for half of the servers. The other half of the servers adds another extra 10 seconds per database to complete one operation. For every Database on a slow server that does have a proper Administration Server setting (a total of 160 000 * 80 / 100 = 128,000) the following 16 seconds has to be spent:

e. Open database (10 seconds)
f. Open ACL (4 seconds)
g. Click on Advanced tab (1 second)
h. Verify the Administration setting (1 second)

This totals to an amount of 704,000 seconds from fast servers and 1,024,000 seconds from the slow servers! The total, 1 728 000 seconds equals to 480 hours which equals to 60 business days of one person for merely detecting faulty ACLs! If you noticed, there is a 22% increase in duration for detection phase compared to the fast servers (49 days vs. 60 days).

Remember that we are assuming that network speed is reasonably fast and uniform for half of the servers. Therefore, manually updating the Administration Server ACL entry in one Database on a slow server involves the following steps.

i. Open database (10 seconds)
j. Open ACL (4 seconds)
k. Click on Advanced tab (1 second)
l. Click on Server option (1 second)
m. Locate/type the correct Administration Server name (4 seconds)
n. Click on Basics tab (1 seconds)
o. Review/Verify you have entered the data correctly (3 seconds)
p. Click OK and wait for it to take effect (7 seconds)

Performing this very basic ACL operation takes 31 seconds per database – assuming that the administrator makes no mistakes along the way. The total average number of Databases with improper ACL settings on a given server is 400 * 20 / 100 = 80. The total number of Databases with improper ACL settings is 80 * 400 = 32,000. Therefore, total time for fixing the faulty ACLs on slow servers adds up to 31 * 32,000 * 1 / 2 = 496,000 seconds. This equals to 137 hours, which corresponds to 17 days of one person! The total from the fast servers is 21 * 32,000 * 1 / 2 = 336,000 seconds. This equals to 93 hours, which corresponds to 11 days of one person. The total time it takes to fix faulty ACLs on slow and fast servers is11 + 17 = 28 days of one person.

The overall total of detecting and fixing faulty ACLs equals to 60 + 28 = 88 business days of one person!

Note: **With the introduction of Notes/Domino version 6, the ability of setting the Administration server on multiple databases has been somewhat improved**.

However, the best that can be done via the Notes 6 Administration client is selecting multiple databases – so that you can set the Administration Server in their ACL - only if they are in the same folder on the same server. Not to mention that you need decent real-time network speed to access the databases in concern and your workstation will be locked up while the ACL change operation is in progress. The overall result is definitely better than the grim and totally manual scenarios described above, but nowhere near the **minute-long operation of ET!**

-End-