

# How Does Notes Domino 6 Document Locking Feature Work?

## Product documentation

### Abstract

This paper is intended to give a brief overview of document locking, which is a new feature beginning with Notes Domino 6 (ND6). Document locking is a way to dramatically reduce replication/save conflicts, and to otherwise ensure that when one user is editing a document, no one else will try to do so.

### Content

This paper is intended to give a brief overview of document locking, which is a new feature beginning with Notes Domino 6 (ND6).

Document locking is a way to dramatically reduce replication/save conflicts, and to otherwise ensure that when one user is editing a document, no one else will try to do so. If the document is in a database replicated amongst Domino 6 servers that communicate easily with one another, then the feature works very smoothly. If some of the servers are extremely remote, or are in different domains and do not communicate directly with each other, or the system is a mix of R5 and Domino 6, then various problems will ensue.

### How to enable document locking

- Enable the feature called "Allow document locking" on the first tab of the Database Properties dialog box
- The database needs to be ODS 43 (ND6)
- You will need to have an administrative server selected in the ACL
- You do not need to compact the database

### What happens once document locking is enabled

Once enabled, you will find that every time you edit a document, the status bar indicates, "Document successfully locked." Likewise, when saving/closing a document, "Document successfully unlocked" is written to the status bar. This requires no extra steps for your users. Behind the scenes, whenever the document is put in/out of edit mode, two fields are written to: \$Writers and \$WritersDate. The first field is a read/write access names field, the same as an AuthorNames field. The second field is a time/date stamp. These fields are used to let the system know that the document is locked. When unlocked, these fields are blanked out.

Note that you can alternatively lock/unlock a document via the Actions menu (or right-click and choose Lock/Unlock from the pop-up menu). If you lock a document this way, it remains locked until you unlock it. If you lock a document implicitly by simply editing the document, it unlocks automatically when you come out of edit mode.

If the database has no replica copies, then using this feature is trivial. But let us assume that there are replica copies on multiple servers, either multiple single servers or in a clustered environment. In that case, when you try to edit a document, the server makes a call to the administrative server to make sure that this document is not locked. If it is locked, you are given an appropriate error message. If it is not locked, then you are given the lock. For this reason, it is critical to the smooth functioning of this feature that the servers can all communicate with the administrative server easily. If they cannot, or if that communication is slow, then your users will experience a host of problems, from incomplete error messages to an inability to edit/save the document.

Note that the feature can handle local and dial-up access. It gives the user a message to the effect that it will try to synchronize the edits when connected, and will generate a rep/save conflict only if necessary. The user will get an email message with the results of this attempt.

## **What happens in a mixed environment**

- R5 client trying to edit an unlocked document. This is not a problem.
- R5 client trying to edit a locked document. Client generates an error message on save, cannot save edits.
- Web browser trying to edit an unlocked document. This is not a problem.
- Web browser trying to edit a locked document. Rep/Save conflict, if appropriate. No error messages. Note that you could program something into your form, or in a WebQueryOpen agent, to check for \$Writers and give a warning to web users if a value is found in that field.
- ND6 client trying to edit a document on an R5 server. The following error message displays:  
"Server does not support this version of the network protocol."

Note that your ND6 client will have the menu option to Lock/Unlock, but the server is not able to support that call.

In summary, this is a very smooth feature if the database replicates among Domino 6 servers that can easily communicate with each other, and if the users have ND6 clients. If your system is not set up this way, you should do appropriate testing to ensure that you can at least train your users as to what error messages or problems they may encounter.

## **What is the Difference Between Soft Locking and Hard Locking in Lotus Notes?**

### **Technote (FAQ)**

#### **Problem**

What is the difference between soft locking and document locking (sometimes called hard locking) and when are these locking features enabled?

#### **Solution**

#### **Document Locking:**

In Notes/Domino 6.x, a database property feature called "Allow document locking" was introduced. When this property is enabled, any user with Author access or above can lock a document *across all replicas*. This prevents two or more users from making changes to the same document and then replicating - which would create replication conflict documents.

To enable this feature, you can select the database property "Allow document locking" in the basics tab.

#### **Soft Locking:**

Soft locking is also a new feature in Notes/Domino 6, but it was not documented as clearly as document locking. Soft locks are in-memory locks that are enabled when documents are in memory, and are not persistent. The editor takes out a soft lock on a document when it is opened, and then unlocks it when the document is closed. During the period of time when it is soft locked, no one else can edit that document but it only affects the current replica where the user is working; it does not affect other replicas of the database.

Prior to release 6.5.1, soft locks could not be disabled. Beginning in 6.5.1 and 6.0.4, users can add a NOTES.INI parameter EDIT\_NO\_SOFT\_LOCKS=1 to their client's notes.ini.

The EDIT\_NO\_SOFT\_LOCKS=1 ini parameter was added as a workaround for customers who had applications that did not work well with soft locking. The parameter essentially reverts locking behavior back to the R5 style, where soft locking did not exist. This parameter disables all soft locking in all databases accessed by that client. It cannot be enabled on a database by database basis.

At this time, there is no other way to disable soft locking. This parameter can be set programmatically in all clients via LotusScript with the following instructions:

```
Dim session As New NotesSession  
Call session.SetEnvironmentVar( "EDIT_NO_SOFT_LOCKS", "1" , True)
```

The above script can be included in a Hotspot button included in a mail to all users.

An enhancement request, for soft locking be an option that can be enabled or disabled via a database property like document locking, has been submitted to Quality Engineering as SPR# JGON5WQLMA. There are no current plans to address this request.

## Can a document be programmatically checked to see if it is soft-locked

### Problem

In Lotus Notes 6.0, a new feature was introduced that "soft locks" documents which works when document locking is not enabled and locks a document only on the single server where it resides. This prevents two users from editing the same document on the same server at the same time.

Soft-locking goes into effect when a document is opened in Edit mode. In cases where a document is closed while still in Edit mode, the document remains locked.

Is there a way to programmatically check whether a document is soft-locked? If a document is soft-locked, can it be unlocked?

### Solution

There is no way to programmatically determine if a document is soft-locked, nor is there a way to programmatically unlock a document in this condition. These issues were reported to Quality Engineering as SPR# JGON5WQL2Z; there are no plans to address these issues.

### Workarounds:

#### To unlock soft-locked documents via the UI:

When creating a memo based on a Stationery document, create a copy of the document and work with the copy. For other documents not based on Stationery documents, use the following steps to unlock documents:

1. Open the document in Edit mode.
2. Place the document into Read mode (press CTRL+E on the keyboard or select Actions --> Edit from the Notes menu to toggle back to Read mode).

3. Close the document.

### **To disable soft locking:**

In Notes/Domino 6.5.1, an additional Notes.INI parameter was introduced to disable soft locking:  
EDIT\_NO\_SOFT\_LOCKS=1

To programmatically add the new parameter to a user's Notes.INI:

The LotusScript SetEnvironmentVar method (of the NotesSession class) can be used to set or modify environment variables. To add the parameter to a Notes Client's Notes.INI file, the following code should be executed by the user. It could be placed in an agent, button, or in the PostOpen event of the database. After executing the code, restart Notes in order for the parameter to be applied.

```
Dim session As New NotesSession  
Call session.SetEnvironmentVar( "EDIT_NO_SOFT_LOCKS", "1" , True)
```

### **Additional steps to avoid the issue in the future (without disabling the soft-locking feature):**

#### Forms other than Stationery:

The issue can be avoided by making sure to switch a saved document from Edit mode back into Read mode before closing it. To do this, forms can easily be redesigned by an application designer. In the case of a Memo form, a designer can add the following existing code below to the QueryClose event:

```
If Not( Source.IsNewDoc ) And Source.EditMode = True Then
```

```
    Source.EditMode = False
```

```
End If
```

#### Stationery scenario:

A partial fix can be accomplished by adding LotusScript code to the QueryOpenDocument event of the "(Stationery)" view. This will only resolve the issue if you create a memo based on the stationery by opening the stationery document and composing and sending the memo. It will not resolve the issue in cases where the menu is used: New Memo --> Using Stationery.

Code to add to the "(Stationery)" view's QueryOpenDocument event:

```
Dim ws As New NotesUIWorkspace  
Dim session As New NotesSession  
Dim docs As NotesDocumentCollection  
Dim doc As NotesDocument  
Dim newDoc As NotesDocument
```

```
If Not (session.GetEnvironmentString("MailStEd") = "2") Then
```

```
    Set docs = Source.Documents  
    Set doc = docs.GetFirstDocument  
    Set newDoc = New NotesDocument (doc.ParentDatabase)  
    Call doc.CopyAllItems (newDoc)
```

```
Call newDoc.RemoveItem ("IsMailStationery")
Call ws.EditDocument (True,newDoc)
Continue = False
```

End If

## **Opening document or composing memo based on stationery results in error: 'The document is already locked..'**

### **Problem**

In mail files which multiple users have access to, the following error message occurs when a user attempts to edit a draft memo, or create a memo using Personal Stationery:

"The document is already locked by <username>"

For example, User A creates a memo based on a particular Stationery document. When User B attempts to create a memo based on the same stationery the error occurs, indicating that User A has the document locked.

The error can also occur in databases which are not based on a mail template. The issue occurs whether or not the Database Property, "Allow Document Locking" is enabled.

At times the issue will not occur if the first user closes the database or logs out of Notes (either by selecting the F5 key or by a ID lockout set up in the User Preferences).

**Note:** This same error will occur when creating new documents with an Action or Agent that uses LotusScript to create the document. For more information on this issue see the following document:

### **Solution**

This issue is relative to the use of Soft Locking. Soft Locking goes into effect when a document is opened in Edit mode. In cases where a document is closed while still in Edit mode the document is not unlocked.

An additional scenario also exists. The second scenario is related to the fact that new documents created in the back-end (including User Interface triggered LotusScript code) are assigned a NoteID of 0. When two or more users create such documents, each document they create has a NoteID of 0. The soft locking feature locks documents based on the NoteID; so, even though there are two (or more) different documents, the soft locking feature recognizes only that NoteID 0 is already locked.

The issue relative to soft locking and new back-end documents was reported to Quality Engineering as SPR# KHAN5U4NVZ and was fixed in Domino 6.0.4/6.5.2 and 7.0. The fix resolves a number of scenarios where soft locking errors were encountered, including the one involving the use of Stationery in a mail file. Refer to the following document for scenarios involving LotusScript:

In cases where the above fix does not resolve the issue then soft locking can be disabled using a Notes.ini parameter introduced in Notes Domino 6.5.1, 6.0.4 and 7.0.

### **Editor**

SPR# GKAA56ETP6 - Document soft-locking can now be disabled client-side in the editor by setting the ini parameter EDIT\_NO\_SOFT\_LOCKS=1.

## To programmatically add the new parameter to a user's Notes.ini:

The LotusScript SetEnvironmentVar method (of the NotesSession class) can be used to set or modify environment variables. To add the parameter to a Notes Client's Notes.ini the following code would need to be executed by the user. It could be placed in a agent, button or the PostOpen event of the database. After executing the code users will need to restart their Notes client in order for the parameter to be applied.

```
Dim session As New NotesSession  
Call session.SetEnvironmentVar( "EDIT_NO_SOFT_LOCKS", "1" , True)
```

## To Avoid the Issue in the Future

Forms other than Stationery:

The issue can be avoided in the future by making sure to switch a saved document from Edit mode back into Read mode before closing it. Forms can easily be redesigned by an application designer to do this. In the case of the Memo form a designer can add the following below existing code in the QueryClose event:

```
If Not( Source.IsNewDoc ) And Source.EditMode = True Then  
Source.EditMode = False  
End If
```

Stationery scenario:

A partial fix can be accomplished by adding LotusScript code to the QueryOpenDocument event of the "(Stationery)" view. This will only resolve the issue if the one creates a memo based on the stationery by opening the stationery document, and composing and sending the memo. It will not resolve the issue in cases where the menu is used: New Memo -> New Memo -> Using Stationery.

Code to add to the "(Stationery)" view's QueryOpenDocument event:

```
Dim ws As New NotesUIWorkspace  
Dim session As New NotesSession  
Dim docs As NotesDocumentCollection  
Dim doc As NotesDocument  
Dim newDoc As NotesDocument  
  
If Not (session.GetEnvironmentString("MailStEd") = "2") Then  
  
    Set docs = Source.Documents  
    Set doc = docs.GetFirstDocument  
    Set newDoc = New NotesDocument (doc.ParentDatabase)  
    Call doc.CopyAllItems (newDoc)  
    Call newDoc.RemoveItem ("IsMailStationery")  
    Call ws.EditDocument (True,newDoc)  
    Continue = False  
  
End If
```

## Workaround:

If the error occurs while attempting to create a memo based on a Stationery document, then the workaround is to create a copy of the Stationery document and then work with the copy.

For other documents not based on a Stationery document use the following steps to work around the issue:

1. Open the document in Edit mode.
2. Place the document into Read mode (press the CTRL+E keys on the keyboard or select Actions --> Edit

from the Notes menu to toggle back to Read mode).

### 3. Close the document.

In addition to the above workarounds there is the option of enabling document locking and then having the original user, or a user with Manager access, use the Unlock Document action to release it. This methodology has the drawback that additional documents could be locked by users while the property is enabled; and those documents, in turn, would need to be unlocked. To enable document locking for a database select File -> Database -> Properties and select the 'Allow Document Locking' checkbox on the first tab of the InfoBox that pops up. To then unlock a document select it and then select from the menu Actions -> Unlock Document.

The above techniques should also assist the user when attempting to unlock a document in cases where the Notes Client crashed or locked prior to the user unlocking the document. The fact that there is not a specific command or function to unlock Soft Locked documents has been reported to Lotus Software Quality Engineering in SPR MROE5KJT5V.

### **Supporting Information:**

Users can create personal Stationery documents by expanding the Tools folder in the left-hand navigator of the mail file (and the view level) and selecting the Stationery folder. When the Stationery view opens to the right, select the New action button and select "Stationery - Personal".

## **Error: 'The document is already locked...' when using actions or agents to create documents**

### **Problem**

In Lotus Domino, actions or agents that use back-end LotusScript to create documents and then open them in the user interface can result in the following error when multiple users operate them in succession:

"The document is already locked by <username>"

Example code which recreates the issue when two or more users executed:

```
Dim s As New notessession
Dim wk As New notesuiworkspace
Dim db As notesdatabase
Dim doc As notesdocument
Dim uidoc As notesuidocument

Set s = New notessession
Set db = s.CurrentDatabase
Set doc = db.CreateDocument
Set doc = New NotesDocument(db)
doc.form = "Main"

Set uidoc = wk.EditDocument(False, doc)
uidoc.editmode = True
```

This issue is related to (although not exactly the same) as the issue in [Document #1100606](#), "Opening Document or Composing Memo Based on Stationery Results in Error: "The Document is Already Locked ..."

### **Solution**

This issue was reported to Quality Engineering as SPR# KHAN5U4NVZ and was fixed in Domino 6.0.4 and 6.5.2.

SPR# KHAN5U4NVZ - This fix prevents a 'not authorized' error that can occur when archiving with compact -a.

In versions prior to 6.0.4 and 6.5.2, the issue is related to the fact that new documents created in the back-end are assigned a NoteID of 0. When two or more users use the action or agent, each new document they create has a NoteID of 0. The soft locking feature locks documents based on the NoteID, so even though there are two (or more) different documents, the soft locking feature recognizes only that NoteID 0 is already locked.

### **Workarounds:**

The key is to save the document since this generates the NoteID (the NoteID property is read-only, so saving is the only way to generate it).

Two workarounds using Save include:

#### **Back-end:**

```
Set doc = New NotesDocument(db)  
Call doc.save(true, true) 'NEW LINE HERE  
doc.form="Main"
```

#### **Front-end:**

Open the document in Edit mode by the EditDocument method, and then save it right away, rather than opening in Read mode and switching to Edit mode.

**NOTE:** The workaround doesn't work if you use EditDocument to open in Read mode, switch to Edit mode, and then try to save.

Replace the following code:

```
Set uidoc = wk.EditDocument(False, doc)  
uidoc.editmode = True
```

with:

```
Set uidoc = wk.EditDocument(True, doc)  
Call uidoc.save
```

Beyond these workarounds, the only other suggestion is to create the document using the ComposeDocument method in the first place, or to disable soft-locking.