21 CFR Part 11 Requirements for Domino and Notes
Copyright and Trademark Information

Disclaimer

THIS DOCUMENTATION IS PROVIDED FOR REFERENCE PURPOSES ONLY. WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS DOCUMENTATION, THIS DOCUMENTATION IS PROVIDED "AS IS" WITHOUT ANY WARRANTY WHATSOEVER AND TO THE MAXIMUM EXTENT PERMITTED, IBM DISCLAIMS ALL IMPLIED WARRANTIES, INCLUDING WITHOUT LIMITATION THE IMPLIED WARRANTIES OF MERCHANTABILITY, NONINFRINGEMENT AND FITNESS FOR A PARTICULAR PURPOSE, WITH RESPECT TO THE SAME. IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES, INCLUDING WITHOUT LIMITATION, DIRECT, INDIRECT, CONSEQUENTIAL OR INCIDENTAL DAMAGES, ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS DOCUMENTATION OR ANY OTHER DOCUMENTATION. NOTWITHSTANDING ANYTHING TO THE CONTRARY, NOTHING CONTAINED IN THIS DOCUMENTATION OR ANY OTHER DOCUMENTATION IS INTENDED TO, NOR SHALL HAVE THE EFFECT OF, CREATING ANY WARRANTIES OR REPRESENTATIONS FROM IBM (OR ITS SUPPLIERS OR LICENSORS), OR ALTERING THE TERMS AND CONDITIONS OF THE APPLICABLE LICENSE AGREEMENT GOVERNING THE USE OF THIS SOFTWARE.

Copyright

Under the copyright laws, neither the documentation nor the software may be copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine-readable form, in whole or in part, without the prior written consent of IBM, except in the manner described in the documentation or the applicable licensing agreement governing the use of the software.

©Copyright IBM Corporation 2003

All rights reserved. Printed in the United States.

Lotus Software
IBM Software Group
One Rogers Street
Cambridge, MA 02142

US Government Users Restricted Rights - Use, duplication or disclosure restricted by GS ADP Schedule Contract with IBM Corp.

List of Trademarks

Domino, Domino Designer, Lotus, Lotus Notes, LotusScript, and Notes are trademarks or registered trademarks of Lotus Development Corporation and/or IBM Corporation in the United States, other countries, or both. Java, JavaScript, and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

All other trademarks are the property of their respective owners.
## Contents

- Introduction .......................................................... 1
- Sec. 11.1 Scope .......................................................... 2
- Sec. 11.2 Implementation ............................................. 3
- Sec. 11.3 Definitions .................................................. 4
  - Digital signatures .................................................. 5
- Sec. 11.10 Controls for closed systems ............................ 6
  - Digital signatures in Notes applications ......................... 6
  - Version tracking .................................................. 9
  - Best practices for digital signatures ............................ 10
  - Database ACLs .................................................. 11
  - Best practices for database ACLs ................................ 11
  - Best practice for protecting records ........................... 15
  - Limiting access to Domino servers .............................. 16
  - Best practices for Domino server security ..................... 16
  - Limiting access to applications ................................ 17
  - Best practices for securing applications ...................... 17
  - Best practices for securing application design elements .... 17
  - The Domino server log (LOG.NSF) .............................. 18
  - Managing administrator access ................................ 18
  - Best practices for managing administrator access ............ 18
  - Programming in Designer ....................................... 19
  - Best practices for programming in Domino Designer .......... 21
  - Workstation security ........................................... 21
  - Best practices for securing workstations ..................... 21
- Sec. 11.30 Controls for open systems .............................. 23
  - Encrypting records and databases ............................... 23
- Sec. 11.50 Signature manifestations ................................ 25
  - Notes/Domino signatures ....................................... 25
- Sec. 11.70 Signature/record linking ............................... 26
  - Signature protection ........................................... 26
- Sec. 11.100 General requirements .................................. 27
- Sec. 11.200 Electronic signature components and controls ...... 28
  - ID files and passwords ......................................... 28
  - Best practices for ID files and passwords ..................... 28
Sec. 11.300 Controls for identification codes/passwords .................. 30
ID file and password security ............................................. 30
Best practices for ID file and password security ....................... 30
Appendix: References to Notes/Domino documentation .............. 33
Introduction

21 CFR Part 11 is a Food & Drug Administration (FDA) regulation, enacted in 1997, which covers electronic records/electronic signatures when used for FDA regulated activities.

Part 11 describes the process for creating, modifying, maintaining, archiving, retrieving, and transmitting electronic records, and the use of electronic signatures when complying with the Federal Food, Drug, and Cosmetic act or any other Food and Drug Administration regulation. The focus of this regulation is to ensure the integrity, trustworthiness/reliability, traceability, and accountability of a company's record management process. Examples of companies affected by this regulation include those who produce pharmaceuticals, color additives, food additives, livestock feed, medical devices, and who convey food.

Domino provides the tools to help administrators and system/application designers to create applications that meet the compliance requirements. The purpose of this paper is to provide technical guidance to our customers who are building or modifying applications that may be used as part of the compliance process. The focus of this paper is on Domino as an application development platform with the Notes client. The instructions and guidance within this document highlight the many tools and options available within Domino to assist our customers in building compliant and secure applications for their environment. Other technologies such as Web-based access are also available but, when dealing with the Web, developers need to be aware that browsers have fewer capabilities and design options than Notes clients. In particular, browsers do not yet support digital signatures and document-level encryption. These limitations may make things more challenging for the designer of the application.

The intended audiences for this paper are system administrators who manage the systems, and system/application designers who develop applications. We expect that administrators and designers already will be familiar with the 21 CFR 11 requirements and how they apply to their company's procedures.

For additional information on Domino Designer and Domino security, go to the Lotus Developer Domain (formerly Notes.Net) at http://www.lotus.com/ldd.

This paper includes the text of the 21 CFR 11 regulation, identified by lines around the text. Sections 11.1 through 11.3 are informational sections, which are provided to assist in understanding the scope, implementation, and definitions used in the 21 CFR 11 regulation.
Sec. 11.1 Scope

(a) The regulations in this part set forth the criteria under which the agency considers electronic records, electronic signatures, and handwritten signatures executed to electronic records to be trustworthy, reliable, and generally equivalent to paper records and handwritten signatures executed on paper.

(b) This part applies to records in electronic form that are created, modified, maintained, archived, retrieved, or transmitted, under any records requirements set forth in agency regulations. This part also applies to electronic records submitted to the agency under requirements of the Federal Food, Drug, and Cosmetic Act and the Public Health Service Act, even if such records are not specifically identified in agency regulations. However, this part does not apply to paper records that are, or have been, transmitted by electronic means.

(c) Where electronic signatures and their associated electronic records meet the requirements of this part, the agency will consider the electronic signatures to be equivalent to full handwritten signatures, initials, and other general signings as required by agency regulations, unless specifically excepted by regulation(s) effective on or after August 20, 1997.

(d) Electronic records that meet the requirements of this part may be used in lieu of paper records, in accordance with Sec. 11.2, unless paper records are specifically required.

(e) Computer systems (including hardware and software), controls, and attendant documentation maintained under this part shall be readily available for, and subject to, FDA inspection.

This section of the 21 CFR Part 11 regulation is included here for reference.
### Sec. 11.2 Implementation

(a) For records required to be maintained but not submitted to the agency, persons may use electronic records in lieu of paper records or electronic signatures in lieu of traditional signatures, in whole or in part, provided that the requirements of this part are met.

(b) For records submitted to the agency, persons may use electronic records in lieu of paper records or electronic signatures in lieu of traditional signatures, in whole or in part, provided that:

   1. The requirements of this part are met; and
   2. The document or parts of a document to be submitted have been identified in public docket No. 92S-0251 as being the type of submission the agency accepts in electronic form. This docket will identify specifically what types of documents or parts of documents are acceptable for submission in electronic form without paper records and the agency receiving unit(s) (e.g., specific center, office, division, branch) to which such submissions may be made. Documents to agency receiving unit(s) not specified in the public docket will not be considered as official if they are submitted in electronic form; paper forms of such documents will be considered as official and must accompany any electronic records. Persons are expected to consult with the intended agency receiving unit for details on how (e.g., method of transmission, media, file formats, and technical protocols) and whether to proceed with the electronic submission.

This section is included here for reference.
Sec. 11.3 Definitions

(a) The definitions and interpretations of terms contained in section 201 of the act apply to those terms when used in this part.

(b) The following definitions of terms also apply to this part:

(1) Act means the Federal Food, Drug, and Cosmetic Act (secs. 201-903 (21 U.S.C. 321-393)).

(2) Agency means the Food and Drug Administration.

(3) Biometrics means a method of verifying an individual's identity based on measurement of the individual's physical feature(s) or repeatable action(s) where those features and/or actions are both unique to that individual and measurable.

(4) Closed system means an environment in which system access is controlled by persons who are responsible for the content of electronic records that are on the system.

(5) Digital signature means an electronic signature based upon cryptographic methods of originator authentication, computed by using a set of rules and a set of parameters such that the identity of the signer and the integrity of the data can be verified.

(6) Electronic record means any combination of text, graphics, data, audio, pictorial, or other information representation in digital form that is created, modified, maintained, archived, retrieved, or distributed by a computer system.

(7) Electronic signature means a computer data compilation of any symbol or series of symbols executed, adopted, or authorized by an individual to be the legally binding equivalent of the individual’s handwritten signature.

(8) Handwritten signature means the scripted name or legal mark of an individual handwritten by that individual and executed or adopted with the present intention to authenticate a writing in a permanent form. The act of signing with a writing or marking instrument such as a pen or stylus is preserved. The scripted name or legal mark, while conventionally applied to paper, may also be applied to other devices that capture the name or mark.

(9) Open system means an environment in which system access is not controlled by persons who are responsible for the content of electronic records that are on the system.
Digital signatures

In Domino and Notes, digital signatures are secured by cryptographic means. Signatures are created with the private key from a user's Notes ID. The assignment of a digital key in a Notes ID is a unique public/private value. (A Notes ID contains encryption keys and certificates that Domino uses to verify the authenticity of a file.) Signatures are stored in documents along with the public key and a list of certificates from the sender's ID.

For more information, see "Digital signatures in Notes applications" in Section 11.10.
Sec. 11.10 Controls for closed systems

Persons who use closed systems to create, modify, maintain, or transmit electronic records shall employ procedures and controls designed to ensure the authenticity, integrity, and, when appropriate, the confidentiality of electronic records, and to ensure that the signer cannot readily repudiate the signed record as not genuine. Such procedures and controls shall include the following:

(a) Validation of systems to ensure accuracy, reliability, consistent intended performance, and the ability to discern invalid or altered records.

Section 11.10 describes controls for closed systems. Domino has the facilities to work in either a closed system or an open system and meet the requirements. Customers must determine which type of system is appropriate. Closed systems require fewer resources to implement the 21 CFR 11 requirements.

Digital signatures in Notes applications

Lotus Notes/Domino has the ability to track any alterations/changes by dates and digital signatures. Notes uses digital signatures as defined in Section 11.3(5). Additional requirements such as Section 11.100(a) require uniqueness of the name. Section 11.50 "Signature manifestations" requires additions to the record that must be supplied by the application, such as the role of the signer.

The developer and the enterprise must establish the method for ensuring the uniqueness of the name as required by 11.100(a), such as searching existing entries in the directory for duplicates. It is also possible to create programmatically a unique number based on the license number in the Notes ID or a hash of the public key in the ID file.

An electronic signature verifies that the person who originated the data is the author and that no one has tampered with the data. As part of the database design, the database designer can determine whether or not users can sign fields, and whether or not sections of a database can be signed.

Designer combines the data in a signature-enabled field with the private key from the sender's user ID to create a unique electronic signature. Designer stores the signature, along with the public key and the list of certificates from the sender's ID, in the document.

Databases can be designed to enable signing of one or more fields on a form. If the field is in an access-controlled section, the signature applies only to the section and is generated when the document is saved. If the field is not in an access-controlled section, the signature is generated only when the document is mailed.
Using digital signatures in Notes applications

For extra security in Notes applications, you can design forms that attach digital signatures to documents, and to access-controlled sections of the documents. A digital signature is an electronic signature with a cryptographic function, which provides a higher level of security. Digital signatures assure a reader that the writer's identity is genuine and that information has not changed since the user saved the document. A digital signature includes the user's hierarchical name, the name of the certifier of the User ID, and the date the document was saved.

How designers create a form for signing

To automatically sign saved documents, designers set the form property, "Sign Documents that Use This Form."

To sign an access-controlled section when a document is saved, you can create at least one field in the section with the property, "Sign if mailed or saved in section." The signature applies only to the section and is generated when the document is saved. To generate multiple signatures on a form, you can create multiple signature-enabled fields in separate access-controlled sections. Signatures are not generated for sections to which a user does not have access.

How Designer stores and verifies digital signatures

Designer combines the data in a signature-enabled field with the private key from the sender's User ID to create a unique electronic signature. Designer stores the signature, along with the public key and the list of certificates from the sender's ID, in the document.

Storing signatures in documents

A document-level signature ("Sign Documents that Use This Form") applies to the entire document.

If a user with Editor access in the database ACL changes the document, Notes replaces the existing signature with the editor's signature when the document is saved.

An unauthorized change to the document causes verification to fail when the document is opened. When verification fails, the application can use both log and audit events. If the design of the application requires, the application can notify the administrator and/or security officer that verification has failed.

Storing signatures in sections

Instead of signing an entire document, you can sign a section within a document and store an electronic signature with the section.

- For documents with one sign-enabled, access-controlled section, Designer stores the signature with the section.
If a user with Editor access in the database ACL changes a sign-enabled section, Designer replaces the existing signature with the editor's signature when the document is resaved.

If there are several sign-enabled fields in the section, Designer uses data from each sign-enabled field in the section to generate a signature. After saving, a change to a field in the document causes verification to fail when a reader opens the document.

For documents with several sign-enabled, access-controlled sections, Designer stores signatures within each section, making it possible to maintain multiple signatures for a document.

If a user with editor access changes one or more sign-enabled sections, Designer replaces all original section signatures with the newer signature when resaving the document. Designer preserves the existing signatures for sections for which the user has no access.

If there are several sign-enabled fields within a section, data from all the sign-enabled fields in that section is used to generate a signature. A change in any field in the document after saving causes verification to fail when a reader opens the document.

**Example of signature verification**

1. Mary saves a sign-enabled document. Notes uses the private key from Mary's User ID and the sign-enabled field data to create a unique signature. Notes also stores Mary's public key and certificates with the document.
2. David opens the signed document to read it. David must have read access to the document.
3. Notes checks to see if the document was signed. If it was, Notes checks the signature against the data to see if it matches.
4. Notes checks the certificates that came from Mary's ID against David's ID to see if they share a common certifier or cross-certificate in the ID.
5. One of the following occurs:
   - If the signature and data are verified, Notes displays a message indicating who signed it.
   - If the data has been modified, Notes displays a message indicating that the document has been changed or corrupted since Mary saved it. David should not assume that the content of the document is what Mary created.
   - If the signature can't be verified or David and Mary don't share a common certificate, Notes displays a message that the signature can't be verified. David should not assume that Mary created the document.
Version tracking

You can use version tracking to maintain a history of changes to a document. To activate version tracking, you must designate the form used to create documents as a version-tracking form. The following version tracking methods are available:

<table>
<thead>
<tr>
<th><strong>Versioning method</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
</table>
| New versions become responses | Use this when the original document is the most important. The original document is listed first in the view; all successive versions follow. Choose this method if the original document is the focal point of the view, with responses being used for reference.  
When new versions become responses, you can prevent replication or save conflicts in the view. If users on different servers modify and save the main document, their versions are treated as two separate response documents when the databases replicate. The two responses are displayed in the view in chronological order. |
| Prior versions become responses | Use this method when the new version is the most important. The latest version is listed first in the view; previous versions and the original follow. Use this method if the update is the most important or most frequently read document and you want to store older versions as a backup or for historical reference.  
When prior versions become responses, you can't prevent replication or save conflicts. If users on different servers modify and save the main document, the two new versions of the document appear as conflicting main documents when the databases replicate. |
| New versions become siblings | Use this method when all versions have equal importance. The original document is listed first in the view; all successive version follow as additional main documents without introducing the risk of replication or save conflicts.  
This method is also useful when revisions aren't based on a historical or subordinate model — for example, in a form where workgroup members create their own replacement versions of an original document or where the original document is used as a template for each new document.  
This method is most effective when you don't expect every main document to be revised, since it is hard to find updates in a view where many new documents have been created in the updating process. To distinguish a revised document from the original document, add identifying information, such as "New Proposal" or "Revised" to the field that displays in the view. |
The following "Create versions choices" are available:

<table>
<thead>
<tr>
<th>&quot;Create versions&quot; choices</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manual - File, New Version</td>
<td>Manually creates a new version of the document only when the user chooses File - Save As New Version. This option allows the user to choose when to create a new version and when to overwrite the existing document.</td>
</tr>
<tr>
<td>Automatic - File, Save</td>
<td>Automatically creates a new version of a document each time the user saves a document.</td>
</tr>
</tbody>
</table>

Make sure users know that version tracking is active, so that they understand the impact of editing documents. Explain version tracking in the "Using Database" document.

If there is a response hierarchy set up in the database, responses to version-controlled documents display as responses to the original document only.

**Best practices for digital signatures**

- Provide each Notes user with a unique ID file and password. The ID file contains encryption keys and certificates that Domino servers use to verify the authenticity of the file (and, indirectly, the authenticity of the person who has the file). The assignment of a digital key in a Notes ID is a unique public/private value. To provide additional security, Notes/Domino allows the use of a password to encrypt sensitive portions of an ID file.

- If a document requires just one signature (for example, to verify who last modified it), you can design the application so that the document is signed before saving by setting the form property "Sign Documents that Use This Form." The database designer also has the option to sign access-controlled sections within the document. An access-controlled section is signed when the document is saved if (a) at least one field in the section has the property "Signed if mailed or saved in section" and (b) the user saving the document has access to the section.

Where a document requires multiple signatures, there are two approaches that the database designer can take:

1) Maintain multiple versions by setting the "Versioning" form property to "Prior versions become responses" or "New versions become responses." These options force the creation of a new document each time a document is modified. Each document contains the signature of the person who created it. The documents are linked in a response hierarchy, which an auditor can follow, for example, through a properly constructed view.

2) Maintain mutually exclusive access-controlled sections. An access-controlled section is not signed, and any existing signature is maintained, if the user saving the document does not have access to the section. To maintain one version of a document containing multiple signatures, create an access-controlled section for each signature. Allow only the signer to access the section.
The database designer or administrator can create additional protection against unauthorized changes by using the access control list (ACL). See the following section on database ACLs.

**Database ACLs**

Every Domino database has an access control list (ACL) that specifies the level of access that users and servers have to that database. Access levels assigned to users determine the tasks that they can perform in a database, while those assigned to servers determine what information within the database the servers can replicate.

For each database user, you select the access level, user type, and access level privileges for each user or group in a database. For further refinement, the database designer can define roles. A role defines a set of users and/or servers and is used in database design elements or functions to restrict access to those elements or functions.

Every database ACL has a log that shows all changes made to a database ACL. Each entry in the list shows when the change occurred, who made the change, and what changed. Only users who have manager access in the ACL can view the ACL log.

**Best practices for database ACLs**

**Default ACL entries**

- Set -Default- access to "No access." Users and servers receive the access assigned to the -Default- entry if they have not specifically been assigned another access level, either individually or as a member of a group, or from a wildcard entry. Setting Default to No Access limits database access to those users and groups specified in the ACL. (You cannot delete Default from the ACL list.)

- Anonymous access should not be allowed. By default, anonymous is set to no access. (You cannot delete Anonymous from the ACL list.)

- The default access for the user who creates the database (Database creator user name) is Manager. Confirm this is the planned access level for this person before you put the database into production. Generally, database creators are given Designer access so that they can make fixes and improvements to the application.

- When you create a new database, the default access for LocalDomainServers is Manager. The LocalDomainServers group lists the servers in the same domain as the server on which the database is stored, and is provided by default with every Domino Directory. The group should have at least Designer access to allow replication of database design changes across the domain.

**Other ACL entries**

- Disallow wildcard entries.

- To control the changes a database receives from a database replica, add server names to an ACL. To ensure tighter security, use the full hierarchical name of the
server — for example, Server1/Sales/Acme — regardless of whether the name of
the server being added is in a different hierarchical organization than that of the
server that stores the database.

- Assign User types to database ACL entries. For example, assigning the Person
  user type to a name prevents an unauthorized user from creating a Group
document with the same person name, adding his or her name to the group, and
then accessing the database through the group name.

- Make sure that the names of terminated employees are removed from the ACLs
  of all databases in your organization. The adminp process will do this
  automatically for most databases. When an employee's name is removed from the
  Domino Directory, each server in the domain deletes the name from the ACLs of
databases for which it is an administration server.

- Define ACL roles to restrict access to database design elements or functions. For
  example, if you have a database of new product information, you could define a
role called "Designers." If you have certain documents in the database that should
be accessible only to product designers, you can assign that level of access to the
document. You then grant that access, through the "Designer" role, to users by
assigning that role to them in the ACL.

- Make sure that the names of terminated employees are removed from the ACLs
  of all databases in your organization.

- If possible, add new names to existing groups in the ACL rather than listing
  names individually. Consider whether to include new names in any roles
  associated with the database. If the database does not use roles, check whether
there are access lists associated with forms, views, fields, or sections, and if so,
consider whether to include new names in these lists.

- Never list individual IDs for administrators directly in the ACLs of production
databases. Instead, use an "Administrators" group. When administrators leave,
remove their names from this group and add the names of their replacements.

General database security

- As part of the design and management decision, define database access levels
  before putting the database into production.

- A best practice for maintaining maximum database security is to use the server
administration process to keep the ACL up to date. The Administration Process
automatically renames or deletes groups, servers, users, personal views, personal
folders, and private agents, and then updates the Domino Directory and any
database ACLs that have named the server running the Administration Process as
their administration server. This program also updates the Readers and Authors
fields for all documents in a database. You can select an administration server for
the Administration Process in the Access Control List dialog box for single
databases or in the Multi-ACL Management dialog box for multiple databases.
To prevent users whose access levels are Depositor or No Access from using the operating system to copy the database, encrypt the database with the server ID through the local Encryption option. This ensures that the database, even when copied, is illegible to anyone who doesn't have access to the server ID.

Select the "Enforce a consistent Access Control List" setting on a database replica whose server has Manager access to other replicas to keep the access control list the same across all server replicas of a database. However, enforcing a consistent access control list does not provide additional security for local replicas. To keep data in local replicas secure, encrypt the database.

Require users to access a database using a secure SSL connection. Secure Sockets Layer (SSL) is a security protocol that provides communications privacy and authentication for Domino server tasks that operate over TCP/IP. You can also choose to require an SSL connection to a single database or to all databases on a server.

Security at the document and field levels

You can restrict access to documents and fields by using Readers and Authors fields, and you can restrict who can create documents or read them by using create access and read access lists.

Using a Readers field to restrict access to specific documents

To limit access to specific documents created from a form, include a Readers field on the form. A Readers field explicitly lists the users who can read documents created from the form. Without Reader access to a document, a user cannot see the document in a view. For example, to limit access to the results of specific patients' reactions to a specific protocol to members of the Clinical Trials department, list the patient and Clinical Trials staff in a Readers field.

If a form has a read access list, names from the Readers field are added to the access list. Otherwise, the Readers field controls access to documents created from the form.

Entries in a Readers field cannot give a user more access than what is specified in the database access control list (ACL); they can only further restrict access. Users who have been assigned "No Access" to a database in the ACL can never read a document, even if you list them in a Readers field. On the other hand, users with Editor access or above in the ACL can be restricted from reading documents if they aren't included in a Readers field.

Any users who have Editor (or higher) access to the database can read and edit a document if one of the following is true:

- They are listed in the form's Read access list or Readers field.
- The form has no Read access list restrictions or no Readers field.
Using an Authors field to restrict who can edit specific documents

An Authors field works in conjunction with Author access in the database ACL. If you assign users Author access in the ACL, they can read documents in the database but cannot edit their own documents. If you list those users in an Authors field, they can edit documents in the database.

Entries in an Authors field cannot override the database ACL; they can only refine it. Users who have been assigned No Access in an ACL can never edit a document, even if you list them in an Authors field. Users who already have Editor (or higher) access in the ACL are not affected by an Authors field. Authors fields affect only users who have Author access in the ACL.

You must enter the user's full hierarchical name in the Authors field.

- If you manually enter a name in the Authors field, you can enter either the abbreviated form, for example, John Smith/ACME/West, or the canonical form, CN=John Smith/OU=ACME/O=West. The name displays in its abbreviated form.
- If you programmatically enter a name, you must enter the canonical form, CN=John Smith/OU=ACME/O=West.

Access-controlled forms and documents

To restrict access to all or part of a form, and to all documents created from a form, you can create a form read access or a create access list.

Create access list

Use a create access list to limit who can access the form in order to create documents. Limiting who can create documents from a form also shortens the create menu by removing the restricted forms from the menu.

Read access list

Use a read access list to limit who can read documents created from a form. For example, you might use a read access list to restrict access to documents that contain personnel information.

The following people can read a document that has restricted Read access:

- Users assigned Read access in the form access list.
- Users listed in the form's Readers field. Readers field names are added to a document's read access list.
- Users listed in the form's Authors field.

Note When you use a form access list, you restrict access to all or part of a form by setting security parameters that work with the database ACL. The database ACL predominates — only users with access to the database have access to forms within a database. Form security provides an additional measure of access control in conjunction with the database access control list. However, using access-controlled forms is not a true security measure because a user can create a copy of the form and remove the restriction.
Replicating restricted documents

Adding names to a document's read access list or to a Readers field limits access to the users, groups, and servers named in that list or field. Servers that need to replicate this database must be included in the list or field to have Read access. Otherwise, documents that are read-restricted won't replicate.

To prevent printing, forwarding, and copying of documents

You can prevent users from printing, forwarding, or copying documents created with a form. To provide additional security and to prevent users from circumventing this feature, do not allow users to install screen capture programs.

To prevent editing of existing documents

You can prevent users with Author access in the database ACL from editing a field in existing documents by selecting "Security options: Must have at least Editor access to use" from the Field Properties box, Advanced tab, and selecting the check mark.

| (b) The ability to generate accurate and complete copies of records in both human readable and electronic form suitable for inspection, review, and copying by the agency. Persons should contact the agency if there are any questions regarding the ability of the agency to perform such review and copying of the electronic records. |
| (c) Protection of records to enable their accurate and ready retrieval throughout the records retention period. |

Secure storage and record retention policies must be part of a customer-defined standard operating procedure.

Best practice for protecting records

Protect records in a database from deletion by disallowing the access level privilege "Delete documents" for all database users.

For information on the use of database ACLs to secure an application, see Section 11.10 (a).

| (d) Limiting system access to authorized individuals. |

Secure storage and record retention policies must be part of the customer-defined standard operating procedure.

The database designer can programmatically hash the contents of a record to determine if the content has changed. For example, the designer might include a "Computed for display" field that applies @Password to a concatenation of all the fields on the form for which you want to detect changes. This "checksum," which can appear on the printed document as well as the online document, can be compared on different versions of a document to ensure that the fields have not changed.
Limiting access to Domino servers

The Domino server is the most critical resource to secure, and is the first level of security that Domino enforces after a user or server gains access to the server on the network.

Best practices for Domino server security

- Cross-certify Notes user IDs and Domino server and certifier IDs so that Notes users and Domino servers in different hierarchically certified organizations can ascertain the identity of users and servers in other Notes organizations.

- Use the Allow Access and Deny Access fields on the Domino Server document to specify which Notes users and Domino servers are authorized to access the server.

- Limit the number of users who can create databases, replicas, and templates on a Domino server to restrict access to the server, and to prevent a proliferation of databases and replicas on the server.

- Control access to a server's network port by granting access only to specific Notes users and Domino servers.

- Encrypt data sent from the server network port to prevent network eavesdropping.

- Password-protect the server console to prevent unauthorized users from entering commands at the server console.

- Secure the server console with a Smartcard to prevent unauthorized access.

- Restrict administrator access by assigning different types of administrator access to individuals based on the tasks they need to perform on the Domino server.

- Specify which Notes users and Domino servers can access the server as a passthru server and specify the destinations they may access.

- Secure the server with SSL, which enables intranet users to authenticate with the server, encrypt data, prevent message tampering, and (optionally) enables the server to use SSL certificates to authenticate clients.

- In multiple server environments, be careful about who is put on the trusted server list. Trusted servers are listed in the Domino Server document; they are trusted to assert the identities of users to the current server and thus are trusted by the current server to have authenticated those users.

- Assign multiple passwords to server and certifier IDs to prevent one user from controlling a server or certifier ID.

- Compare Domino public keys in user IDs with the public key stored in the Domino Directory to prevent unauthorized users from using illicitly obtained IDs to authenticate with Domino.
Limiting access to applications

After users and servers gain access to a server, you can use the database access control list (ACL) to restrict the access that specific users and servers have to individual applications on the server.

For information and best practices on the use of database ACLs to secure an application, see Section 11.10 (a)

Best practices for securing applications

- Encrypt applications to prevent unauthorized users from accessing an application locally on a server or workstation, as ACL settings protect server databases, not locally-stored databases.

- Sign databases and templates to identify their creators. When users access the application, the signature is checked to determine whether the action is allowed on the user workstation.

Best practices for securing application design elements

Securing application design elements prevents users with access to applications from accessing specific design elements in the application -- for example, forms, views, and folders. Application design security takes effect after users gain access to an application.

- Create Read access lists for views to specify which Notes users can see a view.

- Create Read and Edit access lists for folders to specify which Notes users can see a folder or update its contents.

- Create Read and Edit access lists for forms to specify which Notes users can create, modify or read documents created with a form.

- Create Readers and Authors fields to specify which Notes users can create, modify, or read documents created with a form.

- Create signed fields to verify that the Notes user who originated the data is the author and that no one has tampered with the data.

- Create encrypted and hidden fields to control which Notes users can access a field in a form.

- Create Read and Edit access lists for sections to specify which Notes users can access sections within documents.

(c) Use of secure, computer-generated, time-stamped audit trails to independently record the date and time of operator entries and actions that create, modify, or delete electronic records. Record changes shall not obscure previously recorded information. Such audit trail documentation shall be retained for a period at least as long as that required for the subject electronic records and shall be available for agency review and copying.
The Domino server log (LOG.NSF)
Every Domino server has a log file (LOG.NSF) that reports all server activity and provides detailed information about databases and users on the server. The log file is created automatically when you start a server for the first time. You can do the following:

- Control the size of the log file
- Record additional information in the log file
- View the log file
- Search the log file

Notes/Domino supports computer-generated time-stamping in audit logs. The Federal Register Vol. 62, No. 54, pg. 13447, comment 74, states that a secure time-stamp is not required. However, Notes/Domino does not allow records to be recorded on a system with a time earlier than the latest time recorded.

The log data can be retained for any required period.

Access to the log can be controlled by the use of user IDs and passwords.

Managing administrator access
Administrators can use the Domino Administrator to specify various access levels for different types of administrators in the organization. For example, you may want to give only a few people system administrator access, while all of the administrators on your team are designated as database administrators.

Domino 6 offers a new level of administrator access — full access administrator. It is the highest level of administrative access to the server and it eliminates the need to run a Notes client locally on a server.

Best practices for managing administrator access
Grant the full access administrator privilege judiciously.

- Create a special Full Access Admin ID file — for example, "Full Access Admin/Sales/Acme" — and put only that name in the Full Access Admin field. You must then either log in with or switch to this user ID to gain this level of access.

- The Full Access Admin ID should be set up to require that multiple passwords be supplied by more than one administrator.

- Create an OU-level certifier for granting full administrator access, and issue additional IDs to trusted administrators — for example, Jane Admin/Full Access Admin/Acme.
• Leave the Full Access Administrator field empty. Add the name of a trusted individual for emergency situations, and remove it when the situation has been resolved.

• Populate the Full Access Administrator field with a limited set of trusted administrators.

• Audit how the full access administrator access privilege is used:
  • Configure the Event Handler to send notification through EVENTS4.NSF when full access administration privileges are invoked.
  • Any database activity performed by an administrator using full access administrator access is recorded in the database activity log, under Database Properties.
  • Use of the feature is logged by the server.

(f) Use of operational system checks to enforce permitted sequencing of steps and events, as appropriate.

Programming in Designer

Formula, LotusScript, Java, and JavaScript code provide an integral programming interface to Lotus Domino Designer. You can attach code to various design elements depending on need. For example, if you create a computed field on a form, you can attach a formula to compute the value of the field. Or you can attach JavaScript code to the onFocus event of a field, which would execute whenever a user places focus on the field. You can also create a formula, LotusScript or Java agent to automatically update all the documents in a database at scheduled times. Non-programming handlers are also available.

Simple action, Formula, LotusScript, JavaScript, and Java code execute in response to the occurrence of events in the following objects: actions, action hotspots, agents, hotspot buttons, databases, fields, folders, forms, formula pop-ups, outlines, pages, script libraries, subforms, and views. For more information on event handlers, see the Programming Guide, Volume 1: Overview and Formula Language.

The LotusScript/COM/OLE interface includes the following:

• NotesACL class

• NotesACLEntry class

• NotesLog class

• NotesDatabase properties and methods: ACL, ACLActivityLog, CurrentAccessLevel, IsCurrentAccessPublicReader, IsCurrentAccessPublicWriter, GrantAccess, QueryAccess, QueryAccessRoles, Revoke Access, Sign
Sec. 11.10 Controls for closed systems

- NotesItem properties and methods: IsAuthors, IsEncrypted, IsReaders, IsSigned, LastModified
- NotesSession methods: HashPassword, VerifyPassword

The Java/CORBA interface includes the following:

- ACL class
-ACLEntry class
- Log class
- Database methods: getACL, getCurrentAccessLevel, grantAccess, queryAccess, queryAccessRoles, revokeAccess
- Document methods: getAuthors, getEncryptionKeys, setEncryptionKeys, isResponse, isSigned, getLastModified, getParentDocumentUNID, getResponses, getSigner, getVerifier, encrypt, sign
- Item methods: isAuthors, setAuthors, isEncrypted, setEncrypted, isReaders, setReaders, isSigned, setSigned, getLastModified

The Notes/Domino formula language includes the following:

- @Author
- @Certificate
- @HashPassword
- @InheritedDocumentUniqueID
- @IsResponseDoc
- @Modified
- @Password
- @Responses
- @UserAccess
- @UserNamesList
- @UserRoles
- @VerifyPassword
Use of authority checks to ensure that only authorized individuals can use the system, electronically sign a record, access the operation or computer system input or output device, alter a record, or perform the operation at hand.

Best practices for programming in Domino Designer

- Use the Domino Administrator client to determine server access rights. For more information, see Section 11.10 (d)

- Disallow, by default, the ability to "Create personal agents" and "Create LotusScript/Java agents" in database and database template ACLs. For information on the use of database ACLs to secure an application, see Section 11.10 (a).

- Disallow the use of server agents — that is, agents that are to run on server databases — in the Domino Server record.

Use of device (e.g., terminal) checks to determine, as appropriate, the validity of the source of data input or operational instruction.

This is a customer-defined standard operating procedure.

Workstation security

An important aspect of workstation security is the operating system. The customer should have standard operating procedures in place to enable and deploy secure features of the operating system.

Notes and Domino offer several features that, when implemented, protect workstation devices from unauthorized access and prevent active content from changing applications and data that exist on a workstation.

Best practices for securing workstations

- Require a password for all Notes user and server IDs to prevent unauthorized users from using illicitly obtained IDs to authenticate with Domino.

- Enforce password quality testing for IDs to prevent unauthorized users from guessing passwords.

- Require users to change their passwords periodically to prevent unauthorized users from using illicitly obtained IDs to authenticate with Domino.

- Lock the user ID after a pre-determined period of inactivity. This feature automatically logs off the ID from the server and prevents unauthorized users from using the workstation.

- Set up ID recovery for user IDs, so that users can regain access to their ID files instead of having to have new IDs issued.
Sec. 11.10 Controls for closed systems

- Provide two-factor authentication for Notes IDs by installing Smartcard readers on user workstations and requiring users to log in to Notes with Smartcards.

- Request that users use F5 to immediately log off from servers to prevent unauthorized users from using the workstation.

- Implement secure execution control lists (ECLs) on all workstations. ECLs protect user workstations against active content from unknown or suspect sources, and can be configured to limit the action of any active content that does run on workstations. The ECL determines whether the signer of the code is allowed to run the code on a given workstation, and defines the access that the code has to various workstation functions. For example, an ECL can prevent another person's code from running on a computer and damaging or erasing data.

- Set up a security settings policy document to manage workstation ECLs and Notes password properties, such as expiration settings, on an organizational level.

(i) Determination that persons who develop, maintain, or use electronic record/electronic signature systems have the education, training, and experience to perform their assigned tasks.

This is a customer-defined standard operating procedure.

(j) The establishment of, and adherence to, written policies that hold individuals accountable and responsible for actions initiated under their electronic signatures, in order to deter record and signature falsification.

This is a customer-defined standard operating procedure.

(k) Use of appropriate controls over systems documentation including:
   (1) Adequate controls over the distribution of, access to, and use of documentation for system operation and maintenance.
   (2) Revision and change control procedures to maintain an audit trail that documents time-sequenced development and modification of systems documentation.

This is a customer-defined standard operating procedure.
Persons who use open systems to create, modify, maintain, or transmit electronic records shall employ procedures and controls designed to ensure the authenticity, integrity, and, as appropriate, the confidentiality of electronic records from the point of their creation to the point of their receipt. Such procedures and controls shall include those identified in Sec. 11.10, as appropriate, and additional measures such as document encryption and use of appropriate digital signature standards to ensure, as necessary under the circumstances, record authenticity, integrity, and confidentiality.

Encrypting records and databases
Notes/Domino supports both digital signatures and encryption of records and databases.

Encrypting documents and fields
A document is considered to be encrypted if it is created from a form that contains one or more encrypted fields. Each encrypted field is linked to a key that encrypts the contents of the field. An encryption key can be secret — that is, a key that you must send to users in order for them to decrypt a field — or public — that is, a key that is already in a user's ID file and in the user's Person document where it is publicly available.

Public key and secret key encryption
Notes uses public key encryption for electronic mail, and Domino Designer also lets you use public key encryption for encrypting fields in documents. Every user has a unique public key associated with their user name and stored in their user ID. Applications reference the keys by the users’ names in a special field called PublicEncryptionKeys. When a document is saved, all the user names in this field are located in the Domino Directory or the user’s personal address book, the corresponding keys are retrieved, and all fields marked with a special property are encrypted with those keys.

Domino Designer also supports secret key encryption that you can use for encrypting fields in documents. You can create and name secret keys and then distribute the secret keys to users so that they can decrypt the protected data. Secret keys, like public keys, are stored in a user's ID. Applications reference the keys by their names in a special field called SecretEncryptionKeys. When a document is saved, the keys named in this field are retrieved from the user’s ID file, and all fields marked with a special property are encrypted with those keys.

Caution Both public and secret keys are stored in the user ID file. Users should securely back up their ID file each time they add a key.

Document encryption
If you are planning to use secret encryption keys rather than encrypting with a public key, create the secret key before you encrypt a document.
You can encrypt documents with keys in several ways:

- Using public keys. You can encrypt documents with public keys on IDs so that only users with those IDs can read the documents. To do this, you enter one or more names in the Public Encryption keys field on the Security tab in the Document Properties box.

- Using a form property. Database designers can use a form property to add one or more keys to a form. Every document created with the form will be encrypted using the encryption keys.

- Using the Database/Document Properties box. Users can use the Database/Document Properties box to encrypt one or more documents with their own encryption keys stored in their ID files. To use the properties box to encrypt documents, the form must contain a field that can be encrypted.

- Using the SecretEncryptionKeys field. The SecretEncryptionKeys field can contain the name of a key, which is automatically used to encrypt documents, or the field can be left blank, allowing users to assign the encryption key. In either case, users must have the key in their ID file to encrypt a field with a secret key.

- You can set up forms with text or keyword fields that allow the user to choose whether to encrypt a document. Designers can also hide the SecretEncryptionKeys field so that users cannot see the names of the encryption keys.

Field encryption
A database designer can encrypt fields with secret encryption keys. To decrypt these fields, users must merge the secret encryption keys into their ID files. If the user does not have the required encryption key, the encrypted fields appear blank.

For information on the use of digital signatures, see Section 11.10 (a).
Sec. 11.50 Signature manifestations

(a) Signed electronic records shall contain information associated with the signing that clearly indicates all of the following:

1. The printed name of the signer;
2. The date and time when the signature was executed; and
3. The meaning (such as review, approval, responsibility, or authorship) associated with the signature.

(b) The items identified in paragraphs (a)(1), (a)(2), and (a)(3) of this section shall be subject to the same controls as for electronic records and shall be included as part of any human readable form of the electronic record (such as electronic display or printout).

Notes/Domino signatures

A Notes/Domino signature contains the hierarchical name of the signer. The database designer and administrator must establish procedures to ensure that a manifestation of the name is unique. Possible procedures include:

- The administrator ensures that every Notes ID contains a unique name.
- The designer programmatically looks up a unique value, such as an identification or license number, and appends it to the displayed name.
- The designer programmatically gets the public key of the Notes ID, hashes it, and appends it to the displayed name.

A Notes/Domino signature contains the date and time of signing.

A Notes/Domino signature does not contain semantics. For versioning, the document can be designed to provide for writing the semantics into the document being saved. For accessed-controlled sections, the section can contain the semantics. In these cases, the designer should provide a limited number of options when the signature is applied. Alternatively, the Notes ID can be associated with a specific role.

See Section 11.10 (a) for more information.
Sec. 11.70 Signature/record linking

Electronic signatures and handwritten signatures executed to electronic records shall be linked to their respective electronic records to ensure that the signatures cannot be excised, copied, or otherwise transferred to falsify an electronic record by ordinary means.

Signature protection
The product is built based on cryptographic best practices for signatures. This implementation links the signature with the digital bits in the document. A change of even one bit in the document signed invalidates the signature. The product provides no ability to cut and paste signatures. The cryptography of the signature would fail even if someone somehow managed to copy a signature.

For information on the use of digital signatures to secure an application, see Section 11.10 (a).
Sec. 11.100 General requirements

(a) Each electronic signature shall be unique to one individual and shall not be reused by, or reassigned to, anyone else.

The assignment of a digital key in a Notes ID is a unique public/private value. The customer policy must have processes to limit this to a single individual.

For all types of encryption except network port encryption, Domino uses public and private keys so that data encrypted by one of the keys can be decrypted only by the other. The public and private keys are mathematically related and uniquely identify the user. Both are stored in the ID file. Within the ID file, the public key is stored in a certificate, but the private key is stored separately from the certificate. The certificate containing the public key is also stored in the Domino Directory, where it is available to other users. The Notes public key is used to encrypt fields, documents, databases, and messages sent to other Notes users, while the Notes private key is used for decryption.

(b) Before an organization establishes, assigns, certifies, or otherwise sanctions an individual's electronic signature, or any element of such electronic signature, the organization shall verify the identity of the individual.

This is a customer policy requirement.

(c) Persons using electronic signatures shall, prior to or at the time of such use, certify to the agency that the electronic signatures in their system, used on or after August 20, 1997, are intended to be the legally binding equivalent of traditional handwritten signatures.

   (1) The certification shall be submitted in paper form and signed with a traditional handwritten signature, to the Office of Regional Operations (HFC-100), 5600 Fishers Lane, Rockville, MD 20857.

   (2) Persons using electronic signatures shall, upon agency request, provide additional certification or testimony that a specific electronic signature is the legally binding equivalent of the signer's handwritten signature.

This is a customer policy requirement.
Sec. 11.200 Electronic signature components and controls

(a) Electronic signatures that are not based upon biometrics shall:
   (1) Employ at least two distinct identification components such as an identification code and password.

Refer to Section 11.100 (a) for more information.

ID files and passwords

(i) When an individual executes a series of signings during a single, continuous period of controlled system access, the first signing shall be executed using all electronic signature components; subsequent signings shall be executed using at least one electronic signature component that is only executable by, and designed to be used only by, the individual.

Lotus Notes/Domino uses an ID file and password to establish a unique identification.

Best practices for ID files and passwords

Notes/Domino allows for additional controls to be incorporated into database design when required.

Specify the level of password quality you want enforced for the ID when registering user or server IDs or creating certifier IDs. The higher the level, the more complex the password and, therefore, the more difficult it is for an unauthorized user to guess the password. Password quality level is enforced when users enter a password for new IDs or when users change the password for an existing ID.

- For optimal security, specify a password quality level of at least 8.

- Have users choose passwords that consist of random, alphanumeric strings that include mixed uppercase and lowercase letters, numbers, and punctuation. An entire phrase is better than a single word because it is easy to remember, difficult to guess, and generally longer than a single-word password.

- Include a misspelled word in phrases to make it more difficult for attackers to guess at the phrase.

(ii) When an individual executes one or more signings not performed during a single, continuous period of controlled system access, each signing shall be executed using all of the electronic signature components.
(2) Be used only by their genuine owners; and

This is a customer policy and training requirement.

(3) Be administered and executed to ensure that attempted use of an individual's electronic signature by anyone other than its genuine owner requires collaboration of two or more individuals.

To provide tighter security for certifier and server IDs, assign multiple passwords to those IDs. Using multiple passwords requires that a group of administrators, rather than one person, work together to access an ID. You also can specify that only a subset of the assigned passwords be required to access the ID.

(b) Electronic signatures based upon biometrics shall be designed to ensure that they cannot be used by anyone other than their genuine owners.

This requirement does not apply to Notes/Domino. Notes/Domino does not distinguish between devices used (for example, biometric devices, smartcards, etc.). Password recommendations apply independently of access method.
Sec. 11.300 Controls for identification codes/passwords

Persons who use electronic signatures based upon use of identification codes in combination with passwords shall employ controls to ensure their security and integrity. Such controls shall include:

(a) Maintaining the uniqueness of each combined identification code and password, such that no two individuals have the same combination of identification code and password.

ID file and password security

The uniqueness of the name can be displayed using a customer policy for uniqueness. This could be a hash of the public key, license number, or a customer employee number.

Notes client/server authentication for NRPC using an ID file is a cryptographic authentication based on RSA public/private keys and certificates. The certificates are not X.509; they are a Notes format that is equivalent from a security perspective. The user ID is listed in the Domino Directory and utilizes the unique public/private key pair.

(b) Ensuring that identification code and password issuances are periodically checked, recalled, or revised (e.g., to cover such events as password aging).

The Domino Administrator provides options for password checking, expiration of passwords, etc. For more information, see the chapter "Securing and Managing Notes IDs" in Administering the Domino System, Vol. 2.

Best practices for ID file and password security

For information on password security, see Section 11.10 (h).

(c) Following loss management procedures to electronically deauthorize lost, stolen, missing, or otherwise potentially compromised tokens, cards, and other devices that bear or generate identification code or password information, and to issue temporary or permanent replacements using suitable, rigorous controls.

When an identification token (a Notes ID) is deauthorized, the public key is removed from the Domino Directory, which prevents a user from using the ID file. The document with valid signatures remains in the system and can still be audited. This would be done as part of the policy defined by the customer.
(d) Use of transaction safeguards to prevent unauthorized use of passwords and/or identification codes, and to detect and report in an immediate and urgent manner any attempts at their unauthorized use to the system security unit, and, as appropriate, to organizational management.

The administration interface using the password management facility can set a value to ‘lock out’ the ID file on a per-user basis. Messages are logged to the server console when someone tries to access a server to which they do not have access. Administrators can set up Event Handlers to record the occurrence of these events.

For additional information about securing ID files, see Section 11.10 (h).

(e) Initial and periodic testing of devices, such as tokens or cards, that bear or generate identification code or password information to ensure that they function properly and have not been altered in an unauthorized manner.

This is a customer policy requirement.
Appendix: References to Notes/Domino documentation

This table provides references to specific sections in the Notes/Domino documentation that provide additional information on the features discussed in this paper.

<table>
<thead>
<tr>
<th>Section #</th>
<th>Requirement</th>
<th>Feature</th>
<th>Book</th>
<th>Chapter</th>
</tr>
</thead>
<tbody>
<tr>
<td>11.10</td>
<td>Controls for closed systems</td>
<td>Digital signatures</td>
<td>Administering the Domino System</td>
<td>&quot;Encryption and Electronic Signatures&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Application Development with Domino Designer 6</td>
<td></td>
<td>&quot;Application Security&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Version tracking</td>
<td>Application Development with Domino Designer 6</td>
<td>&quot;Designing Forms&quot; &quot;Designing Fields&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Database ACLs</td>
<td>Administering the Domino System</td>
<td>&quot;Controlling Access to Databases&quot;</td>
</tr>
</tbody>
</table>

Use this when the original document is the most important. The original document is listed first in the view; all successive versions follow. Choose this method if the original document is the focal point of the view, with responses being used for reference. When new versions become responses, you can prevent replication or save conflicts in the view. If users on different servers modify and save the main document, their versions are treated as two separate response documents when the databases replicate. The two responses are displayed in the view.
<table>
<thead>
<tr>
<th>Section #</th>
<th>Requirement</th>
<th>Feature</th>
<th>Book</th>
<th>Chapter</th>
</tr>
</thead>
<tbody>
<tr>
<td>34.21</td>
<td>Readers and Authors fields</td>
<td>Application Development with Domino</td>
<td>&quot;Designing Fields&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Designer 6</td>
<td>&quot;Application Security&quot;</td>
<td></td>
</tr>
<tr>
<td>34.22</td>
<td>Access-controlled forms and documents</td>
<td>Application Development with Domino</td>
<td>&quot;Application Security&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Designer 6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>34.23</td>
<td>Controlling access to Domino servers</td>
<td>Administering the Domino System</td>
<td>&quot;Controlling Access to Domino Servers&quot;</td>
<td></td>
</tr>
<tr>
<td>34.24</td>
<td>Managing administrator access</td>
<td>Administering the Domino System</td>
<td>&quot;Controlling Access to Domino Servers&quot;</td>
<td></td>
</tr>
<tr>
<td>34.25</td>
<td>Programming in IBM Lotus Domino Designer</td>
<td>Programming Guide, Volume 1: Overview</td>
<td>&quot;Formula Language&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>and Formula Language</td>
<td>&quot;LotusScript/COM/OLE Classes&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>&quot;Java/CORBA Classes&quot;</td>
<td></td>
</tr>
<tr>
<td>34.26</td>
<td>Workstation security</td>
<td>Administering the Domino System</td>
<td>&quot;Protecting User Workstations with</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Execution Control Lists&quot;</td>
<td></td>
</tr>
<tr>
<td>34.27</td>
<td>Policies</td>
<td>Administering the Domino System</td>
<td>&quot;Using Policies&quot;</td>
<td></td>
</tr>
<tr>
<td>11.30</td>
<td>Controls for open systems</td>
<td>Document and field encryption</td>
<td>Application Development with Domino</td>
<td>&quot;Application Security&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Designer 6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11.50</td>
<td>Signature manifestations</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11.70</td>
<td>Signature/record linking</td>
<td>Use this when the original document is</td>
<td>Use this when the original document is</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>the most important. The original</td>
<td>the most important. The original</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>document is listed first in the view;</td>
<td>document is listed first in the view; all</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>all</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Section #</td>
<td>Requirement</td>
<td>Feature</td>
<td>Book</td>
<td>Chapter</td>
</tr>
<tr>
<td>-----------</td>
<td>----------------------</td>
<td>------------------------------</td>
<td>-------------------------------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>11.100</td>
<td>General requirements</td>
<td>Public and private keys</td>
<td>Administering the Domino System</td>
<td>&quot;Encryption and Electronic Signatures&quot;</td>
</tr>
<tr>
<td>11.300</td>
<td>Controls for identification codes/passwords</td>
<td>ID file and password security</td>
<td>Administering the Domino System</td>
<td>&quot;Securing and Managing Notes IDs&quot;</td>
</tr>
</tbody>
</table>