Summary of Practical Guide to 21 CFR Part 11

NIALL O'ROURKE

PRACTICAL GUIDE TO 21 CFR PART 11

Your Essential Handbook for Navigating 21 CFR Part 11

1st Edition

Niall O'Rourke

Table of Contents

1 Introduction

This PDF summary offers a glimpse into the key topics and insights from Practical Guide to 21 CFR Part 11. While it is designed to provide a taste of the comprehensive content covered in the full book, it also delivers standalone value for professionals navigating the complexities of regulatory compliance.

You'll find concise explanations, practical examples, and actionable tips aimed at bridging the gap between regulatory requirements and real-world implementation. Whether you're new to 21 CFR Part 11 or looking for a refresher, this summary is structured to give you an understanding of foundational concepts, including electronic records, audit trails, and electronic signatures.

In today's highly regulated pharmaceutical industry, ensuring compliance with FDA regulations is more than just a legal obligation—it is a critical pillar of maintaining product quality, patient safety, and operational integrity. Among these regulations, 21 CFR Part 11 plays a pivotal role by establishing the criteria for managing electronic records and electronic signatures in a way that ensures they are as trustworthy and reliable as traditional paper records and handwritten signatures.

As the pharmaceutical industry increasingly adopts digital systems, the challenge lies in bridging the gap between stringent regulatory requirements and the practicalities of real-world implementation. My book, Practical Guide to 21 CFR Part 11, is designed to address that challenge. It provides clear, actionable guidance to help professionals navigate the complexities of compliance with confidence and clarity.

Whether you are a seasoned Validation Engineer, a Quality Assurance specialist, an Automation Engineer, or someone new to the field, this guide offers:

- A comprehensive understanding of the regulatory expectations surrounding electronic records and signatures.
- Practical tools such as checklists, examples, and step-by-step procedures for ensuring compliance.
- Insights into IT and automation technologies essential for compliance in today's digital landscape.

This book goes beyond the theoretical. It is a hands-on resource to help you translate regulations into actionable strategies that work within your organization. By understanding the principles behind the rules and implementing them effectively, you can build systems that not only comply with regulatory requirements but also improve operational efficiency and data integrity.

Embark on this journey through 21 CFR Part 11, and discover how to turn compliance into an opportunity to strengthen your processes, enhance data security, and drive excellence in pharmaceutical manufacturing.

2 What is 21 CFR Part 11?

The Code of Federal Regulations (CFR) is a collection of rules and regulations established by U.S. federal agencies. Title 21 of the CFR specifically pertains to the rules enforced by the Food and Drug Administration (FDA) and other health-related agencies. Among these regulations, 21 CFR Part 11 is particularly significant for industries regulated by the FDA. It governs the use of electronic records and electronic signatures (ERES), ensuring they are as reliable, trustworthy, and legally equivalent to traditional paper records and handwritten signatures.

Purpose of 21 CFR Part 11

The primary objective of 21 CFR Part 11 is to establish the criteria under which electronic records and signatures are considered valid and secure. This regulation enables organizations to move away from paper-based documentation to electronic systems, streamlining processes while maintaining the integrity and authenticity of records required for FDA compliance.

Summary of Key Provisions

The regulation is divided into several key sections:

1. Electronic Records

    - Controls for Closed Systems: These systems, where access is controlled by those responsible for the content, must include stringent controls to ensure the authenticity, integrity, and confidentiality of electronic records.

    - Controls for Open Systems: For systems where access is not restricted to the record's content owner, additional measures like encryption and digital signatures are mandated to protect the records.

    - Audit Trails: Secure, time-stamped audit trails must be maintained to record all user actions that impact electronic records.

    - Record Retention: Records must be preserved and remain accessible throughout their required retention period.

2. Electronic Signatures

    - Unique Signatures: Each electronic signature must be uniquely assigned to an individual and not be reused by others.

    - Verification of Identity: Organizations must verify an individual's identity before they are permitted to execute electronic signatures.

    - Signature Manifestations: Signed electronic records must display the signer's name, the date and time of signing, and the meaning associated with the signature (e.g., approval, review).

Importance of Compliance

Compliance with 21 CFR Part 11 is crucial for organizations within FDA regulated industries. It ensures that electronic records and signatures are held to the same standards of validity and reliability as their paper counterparts. Non-compliance can lead to serious repercussions, including regulatory action, financial penalties, and damage to an organization's reputation.

3 Electronic Records

3.1 Controls for Closed Systems

3.1.1 Overview and What is a Closed System?

According to the 21 CFR Part 11 guidance, a Closed System is defined as:

> "An environment in which system access is controlled by persons who are responsible for the content of electronic records that are on the system."
> Ref.: Subpart A—General Provisions, 11.3 Definitions, (4)

A closed system is a computer system where access is controlled and limited to authorized individuals only, such as employees within a company.

As part of the 21 CFR Part 11 regulation, organizations are required to employ procedures and controls to ensure the authenticity, integrity, and confidentiality of electronic records. Before delving into the specifics of these controls and procedures, here is an extract from the Part 11 guidance:

> "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."
> Ref.: Subpart B—Electronic Records, 11.10 Controls for closed systems, (a)

Below is an overview of some of the key controls for electronic records in a closed system.

For a comprehensive and detailed breakdown of all required controls, refer to my book, Practical Guide to 21 CFR Part 11. The book offers an in-depth exploration of closed system requirements, complete with practical examples and actionable steps to help ensure compliance within your organization.

3.1.2 Generate Accurate and Complete Copies of Records

> "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."
> Ref.: Subpart B—Electronic Records, 11.10 Controls for closed systems, (b)

Identify what records your system procedures, for example:

- Audit Trails
- Batch Reports

It is best practice to export these records in common file formats, such as PDF or CSV, which are widely accessible without proprietary software. If your records are only available in proprietary formats, ensure that the necessary software to open these files is available, and maintain a backup of the software. This backup must be kept for the entire retention period of the records, ensuring ongoing access to the information.

If you are already planning to test alarms or authentication events, you can efficiently test this records requirement by adding a final test step to export the audit trail or alarm and event log.

If you don't have an existing test to incorporate this step, here is a simple example to generate some audit trail data and then export it.

Audit Trail Entry and Export Verification

Step	Procedure	Expected Result
1	Generate an entry in the audit trail. Attach a screenshot of the audit trail. Example: An entry in the audit trail can be generated by triggering an authentication event by logging into the system.	An audit trail entry has been generated. Screenshot attached.
2	Export the audit trail record. Confirm that the entry generated above has been successfully exported as part of the audit trail. Attach the audit trail export.	The entry generated above has been successfully exported as part of the audit trail. The audit trail export has been attached.

Step 1: The screenshot will show the audit trail as viewable on the screen by the user.

Step 2: This step accomplishes two objectives:

1. It confirms that the audit trail entry viewable on the screen is indeed added to the export, ensuring accuracy in compliance with 21 CFR Part 11 requirements.

2. It confirms that the audit trail record can be exported/copied, which also complies with the requirements.

Testing New Systems

If your system is brand new, consider testing that the following information is accurately added to the audit trail:

- Authentication Events: Log in, log out, etc.
- Parameter Changes
- Alarm Generation
- Alarm Acknowledgment

When testing these, you should incorporate a mechanism for verifying that the audit trail is accurate. A practical way to do this is to take screenshots of the audit log on screen as you go through the tests. After each alarm or event is generated, confirm that it is successfully shown in the on-screen audit trail viewer. This allows you to confirm that the audit trail export matches the on-screen records in a later test. If the on-screen audit trail is not accurately recording the information, the test executor should notice this and fail that test.

Taking screenshots also allows reviewers to check that the information shown on the on-screen audit trail viewer matches the exported data at the end. This way, you leverage the review of others, not just relying on the executor.

Summary of Accuracy Verification for Audit Trails

Incorporate screenshots during testing to verify that the audit trail is accurately capturing events. This provides a method to confirm on-screen records match exported data, ensuring comprehensive verification by both executors and reviewers.

3.1.3 Protection of Records

> "Protection of records to enable their accurate and ready retrieval throughout the records retention period."
> Ref.: Subpart B—Electronic Records, 11.10 Controls for closed systems, (c)

One of the most critical aspects of protecting records is ensuring that they are backed up. If data doesn't exist in at least two places, then it effectively doesn't exist. Here are some practical steps to help protect your records:

1. Use RAID (Redundant Array of Independent Disks):

   Purpose: Enhance storage reliability and performance by distributing data across multiple disks.

   Implementation: Configure your server with RAID 1 to mirror data across two disks, ensuring that if one disk fails, the other still has a complete copy.

   Note: RAID is not a substitute for backup. RAID can be configured in software or with RAID cards. You can verify the RAID configuration during the Installation Qualification of the system.

2. Database Backups:

   Purpose: Ensure that your critical data can be restored in case of corruption or loss.

   Implementation: Use tools provided by the database provider (e.g., SQL tools) for regular backups. If no tools are available, manually copy the database periodically.

   Example: Schedule regular full and incremental backups of your database.

3. Operating System Image Backups:

   Purpose: Backup the entire operating system to allow for full system restoration.

   Implementation: Use tools like Acronis to create a complete image of the operating system in a bootable format. This ensures that you can restore the entire system, including all settings and configurations.

   Example: Perform regular Acronis image backups, with differential backups to capture changes since the last full backup.

4. Snapshots to Prevent Overwriting Good Backups:

   Purpose: Ensure that you don't overwrite good backups with corrupted data or data that has been encrypted by ransomware.

   Implementation: Perform regular snapshots of your data at various points in time. Snapshots capture the state of your data at a specific moment, allowing you to restore to a known good state if needed.

   Example: Schedule daily snapshots in addition to your regular backups. In case of data corruption or ransomware, you can restore from the last known good snapshot rather than a potentially corrupted backup.

You need to have documentation that details the backup mechanisms and frequency.

Testing of Backups

Regularly testing backups is essential to ensure that data can be accurately restored when needed, verifying the integrity and reliability of backup processes.

For Database Backups:

- To test database backups, you can simply open the backup file or restore it into a blank database.
- **Example:** Use SQL Server tools to restore a database backup into a new, empty database instance and verify that the data and structure are intact.

For Acronis Image Backups:

- To test Acronis image backups, you need to restore the image onto spare physical hardware or a virtual machine.
- This step is crucial to check that the backup is actually restorable and that the system can boot and function correctly.
- **Example:** Use Acronis tools to restore a full system image to a virtual machine and verify that the operating system boots up and operates as expected.

3.1.4 Limiting System Access to Authorized Individuals

> "Limiting system access to authorized individuals."
> Ref.: Subpart B—Electronic Records, 11.10 Controls for closed systems, (d)

It is essential to limit system access to authorized individuals only. This involves both physical and digital access controls. Here are some practical steps to achieve this:

1. Access to Server Room and Site:

   Implementation: Require badge access to enter the physical building and the server room.

   Example: Use electronic badge systems to control and log access to critical areas, ensuring that only authorized personnel can enter.

2. Physically Locked Server Cabinets:

   Implementation: Ensure that server cabinets are locked to prevent unauthorized physical access.

   Example: Use key or combination locks on server cabinets and restrict access. The keys are often generic and can be bought online, but it still offers a level of protection against opportunistic individuals.

3. Securing Unused USB and Network Ports:

   Implementation: Disable unused USB and network ports in software or physically lock them to prevent unauthorized use.

   Example: Use BIOS/UEFI settings or software policies to disable unused ports and apply physical locks where necessary.

4. User Authentication

   Implementation: Require users to be logged in with a unique username and password to perform any actions on the system.

   Example: When a user is not logged in, the system defaults to a view-only state where users can view information but cannot make any changes. This ensures that only authenticated users can modify the system, enhancing security and accountability.

5. User Access Control:

   Implementation: Develop a user access matrix that defines different levels of access for various roles (e.g., administrator, supervisor, engineer, operator).

   Example:
   - Administrator: Full system access, including configuration and user management.
   - Supervisor: Access to supervisory functions and system monitoring.
   - Engineer: Access to engineering and maintenance functions.
   - Operator: Access to basic operational functions only.

Additional details and testing procedures for the user access matrix will be covered in Section **Error! Reference source not found.**.

Testing:

The following can be checked during the Installation Qualification:

- The server room is restricted to authorized users via badge access.
- Server cabinets are lockable.
- Unused USB and network ports are disabled or physically blocked.

When testing user access, you need to consider the following:

1. Local Users:

   Local users are accounts that are created and managed on the individual system or server. Typically, these are created and managed via the systems application (operator interface)

2. Domain Users:

   Domain users are accounts that are managed centrally by a Windows Domain Controller within a network. These users can be configured to access multiple systems and resources within the domain, based on their permissions.

3. User Groups:

   User groups are collections of user accounts that share the same access rights and permissions. Each user group is configured according to the user access matrix, which defines the levels of access for different roles (e.g., administrator, supervisor, engineer, operator). This ensures that users have appropriate access based on their job functions. You can have local user groups and domain user groups. Typically, they are configured to match.

Let's first test logging in as a member of a particular group for both local and domain users. I'll show an example using a domain user, but the process for a local user is very similar.

Domain User Login

Step	Procedure	Expected Result
1	Log in as a **domain** user who is a member of the "**Operator**" group. Attach a screenshot showing that the user is logged in and that they are a member of the "**Operator**" group.	The user is logged in and they are a member of the "**Operator**" group. Screenshot attached.

You can then repeat the above testing for all user groups, e.g., administrator, supervisor, engineer, etc.

Sometimes, the system application (operator interface) will show the username of the logged-in user and the user group they belong to. In such cases, just take the screenshot and you're done.

If the system application does not show the user group of the logged-in user, one option is to open Active Directory, navigate to the user, check which groups they are a member of, and confirm it that way.

3.1.5 Time-Stamped Audit Trails

> "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."
> Ref.: Subpart B—Electronic Records, 11.10 Controls for closed systems, (e)

All modern systems that have an audit trail are typically configured so that once an audit trail entry is generated, it is fixed and never updated.

Some systems have size limits on the number of audit trail entries they can store. They sometimes implement a first-in, first-out process where the oldest entry is removed to add a new entry without exceeding the size limit of the audit trail.

You need to identify if your system has such a limit and implement a backup and archive of historical entries before the system automatically removes them. This could be a manual activity performed periodically per procedure or configured to run automatically. If it is automatic, I suggest validating this process as you are using it to prevent data loss.

A robust backup strategy is essential to meet the requirement of retaining audit trail documentation for review and copying by auditors. Relying solely on the source machine for data storage carries the risk of unrecoverable data loss due to drive failure, water damage, cyber attacks, and other potential threats.

It is important to understand that an action attributed to a user in the audit trail, such as acknowledging an alarm or changing a parameter, is not the same as the user electronically signing to complete that activity. This topic will be covered in more detail in subsequent sections, but an electronic signature requires the user to enter their password at a minimum to reauthenticate, even if they are currently logged in. This is to prevent someone from being able to electronically sign on their behalf if they walk away from their computer or operator interface without logging out.

Here is a list of common audit trail entries:

- Authentication Events: Logins, logouts, etc.
- Parameter Changes
- Alarm Generation
- Alarm Acknowledgment

When you generate an alarm, you should expect to see an initial audit trail entry documenting the generation of the alarm and at least one more entry documenting its acknowledgment. This occurs when the operator acknowledges the alarm. This is an example of audit trail entries not being updated after they are generated, which is expected behavior. This helps satisfy the requirement that record changes shall not obscure previously recorded information.

The best way to test the audit trail is to incorporate it into the testing you are already performing. For example, if you are testing user logins, authentication events will be generated during those tests. Add a step at the end to export the audit trail and confirm that the authentication events are present. This approach also applies to testing alarm generation and parameter changes.

If you need to create standalone tests for the audit trail, here are a few simple examples:

Audit Trail Testing – User Authentication Events

Step	Procedure	Expected Result
1	Generate a user authentication event and confirm that the following is recorded in the audit trail: • Timestamp • Event Type • User Attach a screenshot of the audit trail viewer from the operator interface showing the above event.	A user authentication event has been generated and the following is recorded in the audit trail. • Timestamp • Event Type • User A screenshot of the audit trail viewer from the operator interface showing the above event has been attached.
2	Attach a copy of the audit trail report/export showing the above authentication event(s).	A copy of the audit trail report/export showing the above authentication event(s) has been attached.

Audit Trail Testing – Specific Events

Step	Procedure	Expected Result
1	Generate the following event types: • Log in • Log out • Password Change • Password Reset	The following event types have been generated: • Log in • Log out • Password Change • Password Reset
2	Confirm that the following is recorded in the audit trail for each of the above events: • Timestamp • Event Type • User Attach a screenshot(s) of the audit trail viewer from the operator interface showing the above events.	The following is recorded in the audit trail for each of the above events: • Timestamp • Event Type • User A screenshot of the audit trail viewer from the operator interface showing the above events has been attached.
3	Attach a copy of the audit trail report/export showing the above authentication events.	A copy of the audit trail report/export showing the above authentication events has been attached.

3.1.5.1 Time Synchronization

The requirement for audit trail entries to be timestamped inherently means that the system time must be accurate.

The best way to ensure this is to synchronize your system time with a reliable time source. This can be done automatically or manually. Additionally, account for daylight saving time if it is applicable in your region. If you choose the manual method, you may need to regularly resync your system time with a known good time source (e.g., every two weeks) depending on the extent of time drift.

Often, you will be dealing with Windows-based systems that are part of a domain. Broadly speaking, you have two options for automatic time synchronization:

1. Point the Windows PC/server to a dedicated Network Time Protocol (NTP) server on your network. The downside of this approach is that if the NTP server goes down for maintenance, you lose your time sync.

2. Set the 'type' parameter under the w32time settings in the registry to NT5DS. This setting allows the system to synchronize its time from the domain controller using the domain hierarchy. This means you don't need to specify an IP address, making the system robust to changes in domain controllers and ensuring continuous and reliable time synchronization.

Both options ensure your system time remains accurate, which is crucial for reliable audit trail entries.

Testing Time Synchronization is Windows

Step	Procedure	Expected Result
1	Confirm the following: • Windows Time Service is running • System is configured to adjust for daylight saving time automatically Attach screenshot(s)	The following has been confirmed: • Windows Time Service is running • System is configured to adjust for daylight saving time automatically Screenshot(s) attached

To check if the Windows Time Service is running, open the Windows Command Prompt and use the command 'sc query w32time'. For detailed information about the service status and the current time synchronization source, run 'w32tm /query /status' in the Command Prompt. To configure the system to automatically adjust for daylight saving time, go to the Date and Time settings in Windows.

3.2 Controls for Open Systems

3.2.1 What is an Open System

According to the 21 CFR Part 11 guidance, an Open System is defined as:

> "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."
> Ref.: Subpart A—General Provisions, 11.3 Definitions, (9)

This means that anyone can create a user account independently, without requiring approval or access permissions from an administrator.

For example, consider a service that allows users to perform digital signatures on PDF documents. In such a system, anyone can sign up and begin creating electronic records without any oversight from the organization responsible for the content on the system.

4 Electronic Signatures

4.1 What is an Electronic Signature?

According to the 21 CFR Part 11 guidance, an electronic signature is defined as:

> "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."
> Ref.: Subpart A—General Provisions, 11.3 Definitions, (7)

Most systems perform electronic signatures using the same username and password that users employ to log in. Here's a breakdown of how this process typically works:

1. **Account Creation:**
    a. When a user is onboarded into a system, they are issued a unique username.
    b. The user then creates a password. This password is not typically stored as human-readable text. Instead, best practice is to hash it. Hashing is a process where the password is transformed into a fixed-length string of characters, which appears random. This hashed value is what gets stored in the database.

    An example of a hash would be:

    a8b6c92f8f7c5c2b84885e4ecb1c31ebf3c9e5c5089eb3dfcd0a42b524c50d0

2. **Signing a Document:**
    a. When a user attempts to sign a document electronically, they are prompted to enter their unique username and password.
    b. The system hashes the entered password and compares it with the stored hash in the database.

3. **Verification and Authentication:**
    a. If the hashed value of the entered password matches the stored hash, the system confirms that the user is authenticated.
    b. Upon successful authentication, the electronic signature is accepted and executed.

In summary, the electronic signature is performed by verifying the hash of the entered password against the hash stored in the database for the given username.

4.2 What is a Digital Signature?

According to the 21 CFR Part 11 guidance, a digital signature is defined as:

> "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."
> Ref.: Subpart A—General Provisions, 11.3 Definitions, (5)

Here is a breakdown of the process of digital signing:

Signer Side:

1. **Data Input:** The process begins with the signer preparing the document or data that needs to be signed.

2. **Hashing:** The document is passed through a hash algorithm. This algorithm generates a fixed-length string of characters, known as a hash, which uniquely represents the data. Hashing is a one-way function, meaning that the original data cannot be reconstructed from the hash.

   An example of what a hash would look like is:

   3f5c2e0e03d201f28d7b3451e8a3c2f4b4c4e1d7d0f8a1b2c3e4f5a6b7c8

3. **Encryption with Private Key:** The signer then encrypts the hash using their private key. This step produces the digital signature. In cryptographic terms, private and public keys are generated in pairs. The private key is kept confidential by the signer, while the public key is shared openly. The encryption process ensures that the digital signature can only be decrypted by the corresponding public key, thereby verifying the authenticity of the signer.

4. **Digitally Signed Document:** The encrypted hash (digital signature) is attached to the document, resulting in a digitally signed document. Although the document may include a visual representation of the signer's signature, this image serves only for visual confirmation. The actual verification relies on the encrypted hash.

5. **Metadata Storage:** The hash, the public key, and the hashing algorithm used are stored in the metadata of the digitally signed document. This allows for future verification without requiring the original signing context.

Verifier Side (Executed when opening the signed PDF):

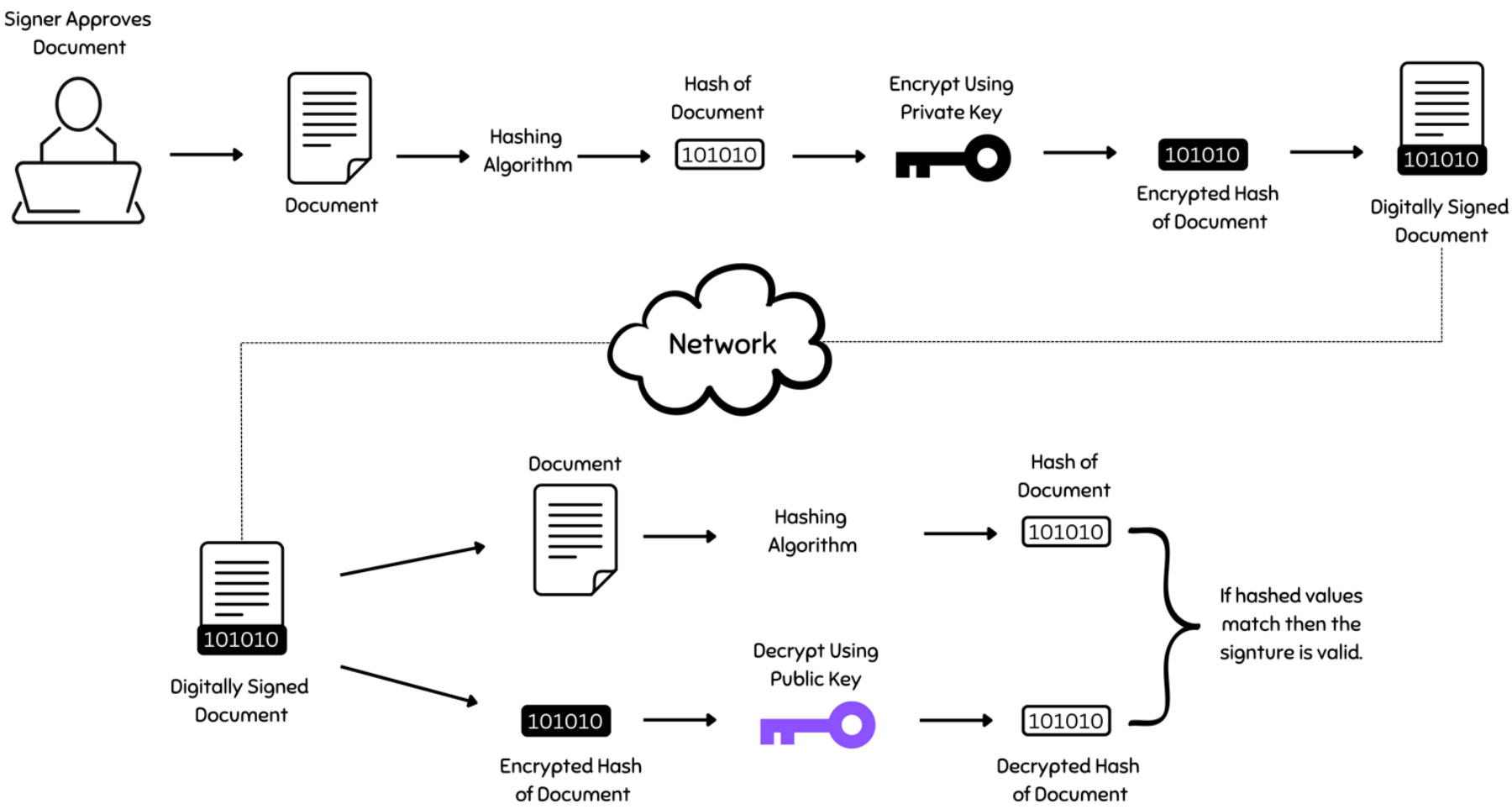
1. **Hashing:** The verifier processes the received document through the same hash algorithm, generating a hash of the document.

2. **Decryption with Public Key:** Using the signer's public key, the verifier decrypts the digital signature, revealing the original hash that was encrypted by the signer. This step ensures that the signature is authentic and that it originated from the signer.

3. **Validation:** The verification process involves comparing the newly generated hash with the decrypted hash. If the hashes match, the signature is considered valid. This confirms that the document has not been altered since it was signed and that it indeed comes from the purported signer. This process is typically automated and occurs when the PDF is opened on a verification site.

4. **Metadata Storage:** The initial encrypted hash and the signer's public key are stored in the PDF's metadata. This storage allows for future verification without requiring the original signing context, ensuring that the document can be validated at any time.

On the next page, you will find a diagram illustrating this workflow.


Signer Approves Document
Document
Hashing Algorithm
Hash of Document
101010
Encrypt Using Private Key
101010
Encrypted Hash of Document
101010
Digitally Signed Document
Network
101010
Digitally Signed Document
Document
Hashing Algorithm
Hash of Document
101010
If hashed values match then the signture is valid.
Decrypt Using Public Key
101010
Encrypted Hash of Document
101010
Decrypted Hash of Document

4.3 Signature Manifest

4.3.1 Essential Elements of Signed Electronic Records

"(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."
Ref.: Subpart C— 11.50 Signature manifestations, (a)(1), (2), (3)

To comply with this requirement, it is essential to confirm that all signed electronic records contain these components:

- The printed name of the signer
- The date and time when the signature was executed
- The meaning (e.g., review, approval etc.)

These components must be present whether the electronic signature is displayed on a screen, such as in a Manufacturing Execution System (MES), or recorded as an event in the audit trail, such as when electronically signing for the acknowledgment of an alarm. Other examples that may require an electronic signature include simulating values or bypassing an interlock.

Step	Procedure	Expected Result
1	Login as an Operator user. Acknowledge an alarm and confirm that the user is prompted to electronically sign for the action. Attach screenshot.	An Operator user is logged in. An alarm has been acknowledged and the user has been prompted to electronically sign for the action. Screenshot attached.
2	Electronically sign for the action.	The action has been electronically signed for.
3	Confirm that the electronic signature record contains the following: 1. The printed name of the signer 2. The date and time when the signature was executed 3. The meaning of the signature Attach screenshot.	The electronic signature record contains the following: 1. The printed name of the signer 2. The date and time when the signature was executed 3. The meaning of the signature Screenshot attached.

4.3.2 Controls for Signed Electronic Records

> "(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)."
> Ref.: Subpart C— 11.50 Signature manifestations, (b)

This section emphasizes that electronic signature records must be treated with the same care and controls as other electronic records. It further specifies that the following elements of the electronic signature must be included in any human-readable records (e.g., reports, exports, printouts):

- The printed name of the signer
- The date and time when the signature was executed
- The meaning of the signature

5 Predicate Rules and 21 CFR Part 11

The Intersection of Predicate Rules and Part 11

When records mandated by other Title 21 requirements (predicate rules) are stored in electronic form, they fall under the scope of 21 CFR Part 11. This means that organizations must comply with both the specific requirements of the predicate rules and the additional provisions of Part 11, which ensures the trustworthiness and reliability of electronic records and signatures.

What Are Predicate Rules?

Predicate rules are the underlying FDA regulations that specify:

- What records must be maintained
- What information those records must contain
- How long records must be retained

These rules are found in various parts of Title 21 of the Code of Federal Regulations (CFR) and apply to FDA-regulated industries such as pharmaceuticals, medical devices, food, and biologics.

Key Points

- Electronic Records of Mandated Information: If you store records required by predicate rules electronically, Part 11 applies.
- Combined Compliance: You must meet the requirements of both the predicate rules and Part 11.

Examples of Predicate Rules

Note: The following examples are accurate at the time of writing. Always refer to the latest FDA regulations and guidance for current requirements.

1. 21 CFR Part 211 – Current Good Manufacturing Practice for Finished Pharmaceuticals

    - Section 211.100: Requires written procedures for production and process control.
    - Section 211.180: Mandates retaining records for at least one year after the batch expiration date.

2. 21 CFR Part 820 – Quality System Regulation for Medical Devices

    - Section 820.30: Requires design control documentation, including verification and validation.
    - Section 820.70: Mandates process control procedures.

3. 21 CFR Part 58 – Good Laboratory Practice for Nonclinical Laboratory Studies

    - Section 58.195: Mandates retention of study records.

4. 21 CFR Part 606 – Current Good Manufacturing Practice for Blood and Blood Components

    - Section 606.100: Requires SOPs for blood collection and processing.
    - Section 606.160: Mandates detailed recordkeeping.

5. 21 CFR Part 312 – Investigational New Drug Application (IND)

    - Section 312.62: Mandates investigators keep subject case histories.

Summary

If you're storing records electronically that are required by predicate rules, they fall under 21 CFR Part 11. It's important to understand how these rules fit together to stay compliant. By combining the requirements of predicate rules with the controls of Part 11, organizations can keep their electronic records and signatures trustworthy and reliable.

6 Definitions

Active Directory: A Microsoft service that manages permissions and access to networked resources, allowing for centralized management of users, computers, and security policies within a domain.

API (Application Programming Interface): A set of routines, protocols, and tools that allow different software systems to communicate with each other, enabling the integration of various software applications.

Backup: The process of copying and storing data to prevent loss in case of system failure, data corruption, or cyberattacks. Backups are essential for data recovery and maintaining system integrity.

Bootable Image: A comprehensive backup method that creates an exact, restorable copy of a computer's entire hard drive, including the operating system, applications, settings, and data. This image can be used to restore the system to its exact state at the time the image was created, making it a crucial tool for disaster recovery.

CAPA (Corrective and Preventive Action): A system used to identify, investigate, and address product and quality issues. CAPA aims to correct problems and prevent their recurrence, ensuring continuous quality improvement.

Checksum: A calculated value used to verify the integrity of data by detecting errors or alterations that may have occurred during transmission or storage.

Closed System: An environment in which system access is controlled by the individuals responsible for the electronic records within the system, ensuring that only authorized users can create, modify, or delete records.

CFR (Code of Federal Regulations): The codification of the general and permanent rules published in the Federal Register by the executive departments and agencies of the U.S. federal government. 21 CFR Part 11 specifically addresses electronic records and electronic signatures in regulated industries.

Domain: In IT, a domain refers to a collection of computers, devices, and resources that are managed under a common set of rules and policies, often within a network.

Domain Controller: A server that manages network security, including user authentication, permissions, and access to resources within a domain.

DIA (Data Integrity Assessment): An evaluation of a system's ability to maintain complete, consistent, and accurate data throughout its lifecycle, ensuring that all data generated by the system is protected, reliable, and readily retrievable for audits and reviews.

Dynamic Data: Electronic records with which the user can interact.

Encryption: The process of converting data into a coded format to prevent unauthorized access, ensuring that only individuals with the correct decryption key can read the data.

Encryption Key: A string of characters used in cryptographic algorithms to encrypt and decrypt data, ensuring that only authorized parties can access the information.

ERES (Electronic Record Electronic Signature): A combined term used in regulated environments to refer to the management of electronic records and electronic signatures, ensuring compliance with regulations like 21 CFR Part 11.

FDS (Functional Design Specification): A document that outlines the functional aspects of a system, detailing how the system will operate to meet user requirements. It typically includes user interactions, control logic, and interfaces.

Hashing: A process that transforms input data into a fixed-length string of characters, known as a hash, which acts as a digital fingerprint for the data. Hashing is commonly used for data verification and security.

HDS (Hardware Design Specification): A document that outlines the hardware components and configurations needed to meet the system's requirements, ensuring that the hardware supports the system's intended functionality.

Network Time Protocol (NTP): A networking protocol for clock synchronization between computer systems, ensuring accurate timekeeping across devices in a network.

Open System: An environment in which system access is not controlled by the individuals responsible for the electronic records, requiring additional controls like encryption and digital signatures to protect record authenticity and integrity.

Predicate Rules: Predicate rules are FDA regulations that specify what records must be maintained, what information they must contain, and how long they must be kept. Found in Title 21 of the CFR, they form the basis for compliance in regulated industries. When stored electronically, these records must also comply with 21 CFR Part 11.

RAID (Redundant Array of Independent Disks): A data storage technology that combines multiple physical disks into a single unit to improve performance and provide redundancy, protecting data against hardware failures.

Rainbow Table Attacks: A type of cyberattack that uses precomputed tables of hash values to quickly reverse-engineer and discover plaintext passwords from their hashed counterparts.

Raw Data: The original records and documentation, preserved in the format in which they were initially created.

Regression Testing: The process of re-testing the system following any changes or updates to ensure that existing functionalities remain unaffected by new modifications or enhancements.

SDS (Software Design Specification): A document that describes the software components, architecture, and other details necessary to meet the system’s functional requirements as defined in the FRS.

Static Data: Records that are fixed and unchangeable, typically in formats like paper or PDF.

User Access Matrix: A chart or table that defines different levels of system access for various roles, specifying what each user group can and cannot do within the system.

PRACTICAL GUIDE TO 21 CFR PART 11

Your Essential Handbook for Navigating 21 CFR Part 11

1st Edition

Niall O'Rourke