PRAXIS
TECHNOLOGY ESCROW

How to Require Software Escrow Protection in Your Contracts

A Comprehensive Guide to Software Escrow, SaaS Escrow, and Technology Risk Mitigation for Enterprise Buyers

Table of Contents

What is Software Escrow? (Explained Simply)

Imagine you run a hospital that depends on special software to track patient records. What happens if the company that makes this software goes out of business? Your hospital would be stuck—you can't access the 'recipe' (source code) to fix bugs or keep the software running.

Software escrow is like having a safety deposit box for that recipe. Here's how it works:

- The software company puts a copy of their source code (the 'recipe') in a secure vault held by a neutral third party
- If something bad happens (the company goes bankrupt, stops supporting the software, or fails to fix critical bugs), you get access to that source code
- With the source code, you can hire another developer to maintain the software and keep your business running

Think of it as an insurance policy for your critical business software. You hope you never need it, but if you do, it can save your company from disaster.

Key Terms and Definition

Understanding these terms will help you navigate software escrow agreements and communicate effectively with vendors and legal teams.

Source Code	The underlying 'recipe' or instructions that make software work. Written by programmers, it can be modified to fix bugs or add features. This is what gets deposited in escrow.
Depositor	The software vendor or SaaS company that creates the software and deposits the source code into escrow.
Beneficiary	Your company, the customer who licenses the software and has the right to access the escrowed materials if release conditions are met.
Escrow Agent	A neutral third-party company that stores the source code securely and releases it only when specific conditions are met. Think of them as the vault keeper.
Release Conditions	Specific events that trigger the release of source code to you (e.g., vendor bankruptcy, failure to provide support, material breach of contract).
Deposit Materials	Everything needed to maintain the software: source code, documentation, build scripts, database schemas, third-party dependencies, and configuration files.

Key Terms and Definition

Verification	The process where the escrow agent confirms that the deposited materials are complete, buildable, and match the current version of the software.
SaaS Escrow	A specialized form of software escrow for cloud-based applications. Includes not just code but also data migration tools, API documentation, and deployment instructions.
Technology Escrow	A broader term encompassing software escrow, source code escrow, and other intellectual property protection mechanisms for critical business technology.
Business Continuity	Your ability to keep operations running even if your software vendor fails. Software escrow is a key business continuity planning tool.
Vendor Risk Management	The practice of identifying and mitigating risks associated with third-party software providers. Requiring escrow is a vendor risk mitigation strategy.
Source Code Access Rights	The specific permissions you have to use, modify, or distribute the escrowed source code once released. These should be clearly defined in your contract.

Why Your Company Needs Software Escrow

Software escrow protects your business from vendor risk and ensures business continuity. Here are the key reasons why enterprise buyers require escrow agreements:

Vendor Bankruptcy Protection

Statistics show that approximately 20% of small businesses fail in their first year, and 50% within five years. If your critical software vendor goes out of business, you could lose access to essential business systems. Software escrow ensures you can continue operating even if your vendor fails.

Mission-Critical System Protection

For systems that are essential to your daily operations—ERP, CRM, manufacturing execution systems, healthcare EMRs, financial trading platforms—downtime is not an option. If your vendor stops supporting the software, you need the ability to maintain it yourself or hire a replacement development team.

Merger & Acquisition Risk

When your software vendor is acquired, priorities shift. The new parent company might discontinue your product, raise prices dramatically, or reduce support. Escrow gives you leverage in these negotiations and ensures continuity if support disappears.

Regulatory Compliance

Many industries—healthcare (HIPAA), finance (SOX, GLBA), government contracting—require business continuity plans for critical systems. Software escrow demonstrates due diligence and risk management to auditors and regulators.

Investment Protection

You've invested significant time and money implementing and customizing software. You've trained employees, migrated data, and integrated systems. Escrow protects that investment by ensuring you're not locked into a failed vendor relationship.

Common Scenarios Where Escrow Saves Companies

A manufacturing company's MES vendor went bankrupt mid-implementation. Escrow allowed them to hire contractors to complete the deployment.

A hospital's EMR vendor was acquired and support was terminated with 90 days notice. Escrow release gave them time to migrate to a new system without disrupting patient care.

A retailer's POS system vendor stopped responding to critical bug reports. The escrow release condition was triggered, allowing them to fix issues in-house.

A financial services firm's custom trading platform developer had a key employee departure that crippled their ability to maintain the code. Escrow provided a safety net for continuity.

How to Require Software Escrow in Your Contracts

Making software escrow a standard requirement in your vendor contracts is straightforward when you follow this step-by-step approach:

Step 1: Identify Which Software Requires Escrow

Not all software purchases need escrow protection. Focus on:

- Mission-critical systems (ERP, CRM, core banking, EMR, manufacturing execution)
- Custom-built or highly customized applications
- Software from vendors with fewer than 100 employees
- Systems that would be expensive or time-consuming to replace
- Applications where you've made significant data migration or integration investments

Step 2: Include Escrow in Your RFP

Make your escrow requirement clear from the beginning:

***Sample RFP Language*:**

"Licensee and the selected vendor shall establish and maintain a three party software / SaaS / Technology escrow agreement with PRAXIS Technology Escrow, LLC. The vendor shall deposit current source code, documentation, and build materials within 30 days of contract execution and provide updated deposits with each major release. The release conditions for the escrow deposit materials shall include but not be limited to: vendor bankruptcy, cessation of business operations, failure to provide contracted support services for 60+ days, or material breach of maintenance obligations."

Step 3: Address Escrow During Contract Negotiation

When negotiating your software license agreement, ensure these points are covered:

1. Who pays escrow fees? (Often vendor pays deposit fees, you pay beneficiary fees, or split costs)
2. Which escrow agent will be used? (You may have a preferred provider)
3. Deposit schedule (automated continuous integration via PRAXIS or similar, OR manual deposits with each release)
4. What's included in deposits (source, docs, dependencies, build scripts)
5. Verification requirements (confirm deposits are complete and buildable)
6. Release conditions (specific events that trigger access)
7. Your rights to use the code after release (maintenance only vs. derivative works)

Step 4: Execute the Escrow Agreement

Typically, there are two separate but related agreements:

- Master Software License / SaaS Subscription Agreement (between you and vendor) - references the escrow requirement
- Escrow Agreement (three-party agreement among you, vendor, and escrow agent) - defines mechanics of deposits and release

Most escrow agents provide standard agreement templates. Review these carefully to ensure they align with your software license agreement. Ensure that the escrow agreement is negotiated and signed along with the License / Subscription agreement to ensure compliance.

Step 5: Monitor Compliance

After establishing escrow, maintain oversight:

- Request deposit confirmation receipts from the escrow agent
- Verify deposits are made with each major software release
- Consider periodic verification testing (especially for critical systems)
- Monitor vendor health (financial reports, news, industry rumors)
- Maintain escrow documentation in your contract management system

Essential Contract Clauses

Use these proven contract clauses as templates for your software agreements. Work with your legal counsel to adapt them to your specific needs.

Basic Escrow Requirement Clause

"Within thirty (30) days of the Effective Date, Vendor shall deposit the Source Code for the Software with [Escrow Agent Name], a neutral third-party escrow agent mutually agreed upon by the parties. Vendor shall update such deposit within thirty (30) days of each Major Release of the Software. Customer shall be named as beneficiary under the escrow agreement with release rights as set forth in Section [X]."

Deposit Materials Definition

"Escrow Deposits shall include: (a) complete and current Source Code in human-readable form; (b) all documentation necessary to install, compile, and build the Software; (c) database schemas and data models; (d) system architecture and design documentation; (e) administrator and user documentation; (f) all third-party software components and licenses required to build and operate the Software; and (g) clear instructions for building the Software from source."

Release Conditions Clause

"The Escrow Agent shall release the Escrow Materials to Customer upon occurrence of any of the following Release Events: (a) Vendor files for bankruptcy or is subject to involuntary bankruptcy proceedings; (b) Vendor ceases business operations or announces its intent to cease operations; (c) Vendor is acquired and the acquiring entity discontinues support for the Software; (d) Vendor fails to provide maintenance and support services as required under this Agreement for a period of sixty (60) consecutive days after written notice from Customer; (e) Vendor materially breaches its obligations under this Agreement and fails to cure within thirty (30) days of written notice; or (f) any other event mutually agreed in writing by Customer and Vendor."

License Rights Upon Release

"Upon release of Escrow Materials to Customer pursuant to a Release Event, Customer shall have a perpetual, irrevocable, non-exclusive license to: (a) use, execute, and display the Source Code; (b) modify and create derivative works of the Source Code solely for Customer's internal business purposes; (c) maintain, correct errors, and make updates to the Software; and (d) engage third parties to perform any of the foregoing activities on Customer's behalf. Customer shall not have the right to distribute, sublicense, or commercialize the Source Code or derivative works thereof."

Verification Rights Clause

"Customer may, at its expense and no more than once per calendar year, engage the Escrow Agent or an independent third party to verify that the Escrow Deposit: (a) includes all materials specified in this Agreement; (b) is complete and in buildable form; and (c) corresponds to the current production version of the Software. Vendor shall cooperate with such verification activities and provide any reasonably requested assistance."

Fee Allocation Clause

"Vendor shall pay all fees associated with establishing the escrow account and making deposits. Customer shall pay all fees associated with its status as beneficiary under the escrow agreement. In the event of a release, Vendor shall pay all associated release fees and costs. If Vendor fails to pay required fees, Customer may pay such fees and offset the amount against payments due to Vendor under this Agreement."

SaaS-Specific Escrow Clause

"For SaaS applications, Escrow Deposits shall additionally include: (a) complete application source code for all tiers (frontend, backend, database); (b) infrastructure-as-code and deployment scripts; (c) API documentation and integration specifications; (d) data export tools and procedures to extract Customer Data in standard formats; (e) instructions for deploying the Software in common cloud environments (AWS, Azure, or Google Cloud); and (f) any proprietary algorithms or business logic necessary to replicate the service functionality."

SaaS Escrow vs. Traditional Software Escrow

Software escrow for cloud-based SaaS applications has unique considerations compared to traditional on-premise software. Understanding these differences is critical for effective risk mitigation.

Key Differences

Aspect	Traditional Software	SaaS Applications
Deployment	Installed on-premise or in customer's data center	Hosted in vendor's cloud environment
What's Escrowed	Application source code, database schemas, documentation	Full application stack, infrastructure code, deployment scripts, data migration tools, API specs
Primary Risk	Loss of ability to update/maintain software	Loss of access to both software AND customer data
Recovery Complexity	Medium - need to build and deploy	High - need to replicate entire cloud infrastructure
Data Concerns	Customer already controls their data	Must include data extraction and migration procedures
Release Urgency	Can continue using existing deployment while planning migration	Immediate - service interruption means total loss of access
Third-Party Dependencies	Relatively few	Many - cloud services, APIs, microservices, CDNs
Testing Feasibility	Can verify build on standard hardware	Requires cloud environment to test deployment

Special Considerations for SaaS Escrow

- **Data Portability: Ensure escrow includes tools and documentation to extract ALL your data in usable formats (CSV, JSON, SQL dumps). Test these extraction procedures before you need them.**
  Your data is often more valuable than the application itself. If the SaaS vendor fails, you need a clear path to retrieve your information.

- **Infrastructure Dependencies: Document all third-party services (AWS, Stripe, Twilio, etc.) and ensure you can replace or configure these yourself.**
  Modern SaaS applications rely on dozens of external services. The escrow should include a complete dependency map and alternative provider options.

- **Deployment Complexity: SaaS escrow should include containerized deployments (Docker/Kubernetes) or detailed infrastructure-as-code (Terraform, CloudFormation).**
  Simply having source code is insufficient. You need to be able to stand up the application in a reasonable timeframe.

- **API and Integration Points: All APIs, webhooks, and integration specifications must be documented so you can maintain connections with your other systems.**
  If the SaaS app is integrated with your CRM, accounting system, or other tools, those integrations must continue working.

Automated Escrow: The Modern Solution for Agile Development

Traditional software escrow was designed for a different era—one where software releases happened quarterly or annually. Modern development practices have rendered these old approaches inadequate. This section explores why automated escrow is essential for today's continuous development environments.

The Problem with Traditional "Push" Escrow

Traditional software escrow relies on vendors manually "pushing" code deposits on a quarterly or release-based schedule. This creates a critical vulnerability: **your escrowed materials are almost always outdated.**

In today's agile development world, software companies deploy updates daily or even hourly. Between escrow deposits, hundreds or thousands of code changes occur—bug fixes, security patches, feature enhancements—none of which are captured in your escrow vault. If a release event occurs, you receive code that may be weeks or months out of date, missing critical fixes and updates your production environment depends on.

Automated Escrow: The Modern Solution

PRAXIS pioneered **Automated Escrow™** to solve this fundamental problem. Instead of relying on vendors to remember to upload code, Automated Escrow connects directly to source code repositories (GitHub, BitBucket, GitLab, Azure DevOps, and others) and continuously captures every commit in real-time.

How Automated Escrow Works

- Direct Integration: PRAXIS connects securely to the vendor's source control system using built in collaborator tools (no new software required)
- Continuous Capture: Every week all connected repositories are cloned and added to the archive
- Complete History: Unlike traditional escrow that replaces the previous deposit, PRAXIS maintains the full git history with **Infinite Retention**
- Verification: Verification services confirm that the escrow deposit materials are complete and functional
- Secure Storage: All materials are encrypted and stored in geographically redundant data centers

Benefits of Infinite Retention

PRAXIS's Infinite Retention policy means nothing is ever deleted from your archive. This creates valuable capabilities beyond basic disaster recovery:

- **Version Rollback:** Access any previous version if a recent update causes problems
- **Forensic Analysis:** Audit code changes over time for compliance, security reviews, or litigation
- **Documentation Archive:** Maintain a complete historical record of specifications, architecture decisions, and technical documentation
- **Regulatory Compliance:** Meet retention requirements for audits and regulatory examinations
- **Knowledge Transfer:** New development teams can study the evolution of the codebase to understand design decisions

Requiring Automated Escrow in Your Contracts

When negotiating your software agreements, specify automated escrow to ensure continuous protection:

Sample Contract Language:

"Vendor shall establish automated source code escrow through PRAXIS or an equivalent provider capable of direct repository integration. The escrow system shall connect to Vendor's source control repositories and continuously capture all commits, branches, tags, and associated documentation without requiring manual vendor deposits. The escrow provider shall maintain infinite retention of all archived materials. Customer shall be named as beneficiary with release rights as specified in Section [X]."

Vendor Objections and How to Address Them

Vendors sometimes resist escrow requirements. Here are common objections and effective responses:

"We're financially stable - you don't need escrow"

Your Response:

"We appreciate your confidence, but business conditions change. Escrow protects both parties - it demonstrates your commitment to long-term support and gives us assurance. Many financially strong companies provide escrow as a standard practice for enterprise customers."

"Escrow is too expensive"

Your Response:

"Escrow costs are minimal compared to our total investment in your solution. Typical fees are $2,000-10,000 annually - a small fraction of our license costs. We view this as essential risk management, similar to insurance. We're willing to share costs or absorb beneficiary fees if that helps."

"Our source code is proprietary/confidential"

Your Response:

"We understand and respect your intellectual property. That's exactly why we use a neutral third-party escrow agent with strict confidentiality obligations. The code remains locked and inaccessible unless specific release conditions are met. We have no interest in your code - we just need assurance of business continuity if something goes wrong."

"We use open source components - escrow doesn't make sense"

Your Response:

"While you may use open source libraries, your business logic, architecture, configurations, and customizations are proprietary. Those components are what we need access to in a continuity scenario. We also need your build procedures, deployment scripts, and integration documentation - none of which are publicly available."

"This is just for startups - we don't need it"

Your Response:

"Actually, even large companies use escrow for enterprise customers. Major vendors like Microsoft, Oracle, and SAP have escrow arrangements for their largest customers. It's an industry best practice for mission-critical systems, regardless of vendor size. The fact that you're established makes escrow easier - you have mature processes for documentation and version control."

"Our SaaS model makes escrow unnecessary"

Your Response:

"SaaS actually increases our need for escrow, not decreases it. With on-premise software, we control our deployment. With SaaS, if you cease operations, we lose access to both the application AND our data immediately. SaaS escrow is specifically designed for this model - it includes data extraction tools and deployment procedures so we can maintain continuity."

Negotiation Tips

- Make escrow a dealbreaker early in negotiations - don't introduce it at the last minute
- Offer to use a vendor they've worked with before to reduce their administrative burden
- Be flexible on fee allocation - sometimes offering to pay beneficiary fees removes friction
- Point to competitors who offer escrow as a standard feature
- Frame it as a mutual benefit - vendors who offer escrow win more enterprise deals
- Get executive buy-in on your side first - CFOs and CIOs typically support escrow for critical systems

Cost Considerations

Understanding escrow costs helps you budget appropriately and negotiate effectively with vendors.

Typical Fee Structure

Fee Type	Typical Cost	Usually Paid By
Initial Setup Fee	$500 - $2,000	Split or Vendor
Annual Depositor Fee	$2,000 - $5,000	Vendor
Annual Beneficiary Fee	$500 - $2,000	Customer
Deposit Update Fee	$300 - $800 per deposit	Vendor
Verification Fee (optional)	$1,500 - $5,000	Customer
Release Fee	$500 - $2,000	Vendor or Customer
Multi-Beneficiary Discount	Lower per-beneficiary cost	Vendor passes savings

ROI Perspective

While escrow has costs, consider the alternatives:

- Cost to replace a mission-critical system: $500,000 - $5,000,000
- Revenue lost during 3-6 month replacement project: Variable but potentially millions
- Legal costs if vendor failure causes business disruption: $100,000+
- **Annual escrow cost: $2,500 - $7,000**

Escrow represents less than 1% of the cost of a vendor failure. For mission-critical systems, it's one of the most cost-effective risk mitigation strategies available.

Checklist for Procurement Teams

Use this checklist to ensure you've covered all essential elements when requiring software escrow:

Pre-Purchase / RFP Stage

- Identified which software purchases require escrow based on criticality
- Included escrow requirement in RFP with clear expectations
- Evaluated vendor responses regarding escrow willingness and experience
- Selected preferred escrow agent (or allowed vendor to propose one)

Contract Negotiation Stage

- Verified escrow requirement is in the master software license agreement
- Confirmed deposit materials definition is comprehensive (source, docs, dependencies, build scripts)
- Negotiated release conditions that are specific and objective
- Defined your rights to use source code after release (maintenance, modification, etc.)
- Agreed on fee allocation (who pays setup, annual, deposit, verification, release fees)
- Established deposit schedule (initial + updates)
- Included verification rights in the agreement
- For SaaS: Confirmed data extraction tools and deployment procedures are included

Escrow Agreement Execution

- Reviewed three-party escrow agreement from the escrow agent
- Verified escrow agreement terms align with software license agreement
- Confirmed release conditions match what was negotiated
- All parties (you, vendor, escrow agent) have executed the agreement
- Received confirmation that initial deposit has been made

Ongoing Monitoring

- Set up process to receive deposit confirmation receipts from escrow agent
- Tracking that deposits occur with each major software release
- Annual fees are being paid (by you and/or vendor as agreed)
- Escrow agreement documents are stored in contract management system
- Monitoring vendor health and market conditions for early warning signs
- Consider periodic verification testing for highest-criticality systems

If Release Becomes Necessary

- Document the release event and gather supporting evidence
- Notify escrow agent formally with release request
- Work with escrow agent through release verification process
- Receive and secure the escrowed materials
- Engage internal or external developers to assess and build the code
- Develop transition plan for maintaining the software going forward

Conclusion

Software escrow is a proven, cost-effective tool for protecting your business from vendor risk. By requiring escrow for mission-critical systems, you ensure business continuity, demonstrate due diligence to stakeholders, and maintain leverage in vendor relationships. The key to successful escrow implementation is making it a standard requirement early in the procurement process, negotiating comprehensive terms that address your specific risks, and maintaining ongoing oversight to ensure compliance.

Whether you're procuring traditional on-premise software or modern SaaS applications, escrow should be a core component of your vendor risk management strategy. The small investment in escrow fees pays for itself many times over by protecting against the potentially devastating consequences of vendor failure.

About PRAXIS Technology Escrow

PRAXIS is revolutionizing software escrow with **Automated Escrow**, a breakthrough technology that connects directly to your source code repositories (GitHub, BitBucket, GitLab, and others) for continuous, real-time escrow deposits.

Why PRAXIS Invented Automated Escrow

Modern software development has changed dramatically. Agile methodologies, continuous integration, and DevOps practices mean that code changes happen constantly—often daily or even hourly. Traditional “push” escrow methods, where vendors manually upload code quarterly or with major releases, virtually guarantee that your escrowed materials are outdated the moment they’re deposited.

PRAXIS solved this problem by pioneering direct integration with source control systems. Our Automated Escrow continuously captures every commit, ensuring your escrowed materials are always current—no vendor action required.

Infinite Retention: Your Complete Software Archive

Unlike traditional escrow services that only maintain the “current” version, PRAXIS offers **Infinite Retention**, nothing is ever deleted. This creates a valuable historical archive of source code, documentation, dependencies, and build artifacts spanning your entire relationship with the vendor.

This historical archive provides benefits beyond disaster recovery. It enables you to roll back to any previous version, audit code changes over time, and maintain a complete forensic record for compliance and litigation purposes.

The PRAXIS Advantage

- Always Current: Direct repository integration means your escrow is updated with every code change
- Zero Vendor Burden: No manual uploads, no vendor compliance issues, no outdated deposits
- Complete History: Infinite Retention provides a full audit trail of all code changes
- Built for Modern Development: Supports agile, DevOps, and continuous delivery workflows
- Compliance Made Easy: Automated verification proves deposits are current and complete

For more information about how PRAXIS Automated Escrow can protect your critical software investments, visit **praxisescrow.com** or contact our enterprise solutions team.

PRAXIS Technology Escrow, LLC

(800) 213 9802 / (770) 459 1202
www.praxisescrow.com or info@praxisescrow.com