

Secure File-Based Transmission in Banking Payments and Reporting

Financial institutions have long relied on file-based data exchange – such as FTP/SFTP and similar protocols – to transmit payment instructions and account reports. This approach, often termed host-to-host connectivity, enables bulk payment files and bank reports to be exchanged directly between corporate systems and bank servers without manual intervention. In both corporate-to-bank and interbank scenarios, secure file transfer channels remain a backbone for moving funds and information, even as real-time APIs emerge. Below, we explore how banks use SFTP and related methods for fund transfer instructions and reporting, the formats and networks involved, examples of bank implementations, security measures, compliance considerations, and how these methods are evolving in the era of APIs and open banking.

Corporate Host-to-Host File Transfers for Bulk Payments

Bulk Payment Files: Large corporations commonly integrate their ERP or treasury systems with banks via host-to-host file transfer. In practice, a company generates a payment file (often containing many transactions) and transmits it to the bank's server over a secure channel (e.g. SFTP). Standardized file formats such as ISO 20022 XML, legacy SWIFT MT messages, and proprietary bank formats are supported to cover various payment types. For example, a corporate might send a single XML file that includes dozens of outgoing payments – salaries, supplier invoices, tax payments, etc. – instead of initiating each transfer individually. The bank's host-to-host service parses this file and routes each instruction to the appropriate payment system or network (ACH, SEPA, SWIFT, wire, etc.) for execution. This bulk-file approach is efficient for high volumes and allows straight-through processing from the client's systems to the bank.

Common File Formats: Banks and corporates have converged on several standard file formats for payments and reporting. The ISO 20022 XML format (e.g. pain.001 for payment initiation, camt.053 for statements) has become the most popular file format for corporate-to-bank integrations globally. Many banks also support legacy SWIFT MT formats; for instance, a file of cross-border wire payments may use SWIFT MT103 messages, or a daily statement file may use the MT940 format. In the United States, the NACHA ACH format is used for batch ACH instructions, and BAI2 is a common format for balance and transaction reports. Virtually every bank can deliver standard statement files like BAI2, MT940, or the newer CAMT reports to clients via SFTP. These files contain rich details on account activity (ACH credits/debits, wire transfers, lockbox deposits, etc.), giving corporates a complete data set for reconciliation. Table 1 below summarizes some prevalent file formats and their uses:

File Format	Description	Use Case
ISO 20022 XML	XML-based standard (e.g. pain., pacs., camt.* messages) for payments and reporting.	SEPA payments, cross-border payments, account statements (global).
SWIFT MT	Legacy SWIFT message formats (MT101, MT103 for payments; MT940/942 for statements).	International wires (MT103), payment requests (MT101), daily statements (MT940).
ACH (NACHA)	ASCII flat-file format for batched ACH transactions in the US.	Domestic ACH credit/debit batches (payroll, direct debits).
BAI2	Bank Administration Institute text format for account reports.	Account balance and transaction reporting (mostly US).
EDIFACT	Legacy EDI file formats (e.g. PAYMUL, FINSTA).	Bulk payments and statements (Europe pre-ISO 20022).

(Table 1: Common file formats used in corporate-to-bank file transfers.)

Transmission Process: Once a corporate payment file is ready, it is sent to the bank over a secure file transfer session. The most common method is Secure File Transfer Protocol (SFTP), essentially a secure SSH-encrypted pipeline into a folder on the bank’s server. The corporate’s treasury system (acting as an SFTP client) either “pushes” the file to the bank, or the bank can “pull” files from the client’s server on a schedule. The connection is authenticated with SSH keys or other credentials, and all data is encrypted in transit. Banks typically support automating these transfers so that files are sent and retrieved at scheduled times or triggered by events, enabling straight-through processing without human involvement. After the file arrives, the bank’s internal payment system validates the file structure and entries. Valid instructions are then processed: for example, domestic payments may be sent to an ACH clearing house in bulk, while international payments are converted into SWIFT messages to be sent via the SWIFT network. Any processing reports (confirmation, error reports, etc.) are placed back on the SFTP server for the corporate to retrieve, along with daily account reports (e.g. end-of-day balances in MT940/BAI2 format).

Example: In the U.S. ACH scenario, a company’s payments for the day might be consolidated and sent to its bank via SFTP by the evening cutoff time. The bank aggregates all clients’ ACH items into a single file and transmits that to the Federal Reserve’s clearing system (using a secure file channel) before the Fed’s deadline. The Fed then processes the batch and routes transactions to receiving banks. In this chain, SFTP-based file transfers occur at multiple stages: corporate to bank, and bank to Federal Reserve. This illustrates how file transfers initiate actual fund movements – the files serve as input to payment networks like ACH. Similarly, for a high-value payment, a corporate might send a file with a wire instruction that the bank’s system uses to initiate a Fedwire or SWIFT payment on the client’s behalf.

Interbank and Clearing House File Exchanges

In addition to corporate integrations, file-based transmissions are used in certain interbank or bank-to-clearinghouse contexts. Many batch payment systems operate by exchanging files of transactions between institutions. For example, the Automated Clearing House (ACH) network in the US involves banks sending batch files to the ACH operator. As noted, an originating bank will send a consolidated ACH file via SFTP (or a Fed-provided secure link) to the Federal Reserve or clearing house each cycle. Likewise, the resulting output files (e.g. return items or settlement reports) are delivered back to banks through secure file transfer channels. This file-based model underpins the high-volume, low-value payments space, where efficiency comes from batching thousands of transactions into one file.

Another example is SWIFT FileAct, a service on the SWIFT network that allows transmission of bulk files securely between institutions. Banks use FileAct to exchange large datasets such as payroll files, large remittance files, or even SEPA payment files between banks/clearing systems. For instance, within the Single Euro Payments Area (SEPA), banks may send bulk credit transfer files to clearinghouses using either SWIFT FileAct or a country-specific file protocol. EBICS (Electronic Banking Internet Communication Standard) is one such file-based protocol widely used in countries like Germany and France for bank-to-bank and corporate-to-bank exchanges. EBICS operates over HTTPS with XML payloads and digital signatures, enabling standardized file submissions for payments and direct debit collections in the SEPA zone. Many European banks and corporates connect via EBICS for domestic fund transfers and reporting, as it provides a unified, secure method to send payment files to any bank that supports the standard. While EBICS and SWIFT FileAct are more specialized networks, they serve a similar purpose: reliably moving batches of transactions or reports between financial entities in a secure, file-oriented manner.

It's worth noting that pure FTP (unencrypted File Transfer Protocol) is no longer used for sensitive interbank data – security is critical, so protocols like SFTP or HTTPS (with encryption) have replaced legacy FTP in practice. Even smaller community banks, if they need to send files to a correspondent bank or receive files from a government agency, will use secure file transfer methods to comply with data protection requirements.

Examples of File-Based Connectivity Platforms

Many global banks offer proprietary host-to-host file transfer services to their corporate clients, often under branded platforms. These services typically support a range of protocols and file formats to accommodate different client needs:

- **CitiConnect for Files (Citibank):** CitiConnect is Citibank's host-to-host channel that allows clients to transmit payment, receivable, and reporting files directly from their ERP/TMS to Citi. It supports a single window for global file exchange, meaning a corporate can send one consolidated file to Citi which then handles distribution to various countries or payment types. CitiConnect for Files accepts ISO 20022 XML, EDIFACT, SAP IDoc, and other formats. On the connectivity side, Citi supports multiple secure protocols: SFTP (with password or SSH key authentication), FTPS (FTP over SSL/TLS), HTTPS posts, and even AS2 (a secure HTTP-based file transfer often used in EDI). Citi often requires PGP encryption on files in addition to the transport security for an extra layer of protection. This combination of SSH transport and PGP file encryption in CitiConnect ensures confidentiality and integrity of bulk payment files.
- **HSBC Connect:** HSBC provides a host-to-host solution called HSBC Connect, which is designed for high-volume corporate transactions. It allows clients to submit large batches of domestic and international payments at once and receive reports back into their ERP. HSBC Connect emphasizes robust security and supports multiple encryption protocols including FTPS, SFTP, HTTPS, PGP, and AS2. In practice, a corporate can choose its preferred protocol (e.g. establishing an SFTP tunnel or sending files via an HTTPS API endpoint) and even layer PGP encryption on the files. HSBC Connect accepts all standard industry file formats – ISO 20022 XML for SEPA or global payments, SWIFT MT formats, local clearing formats, etc. – and will automatically validate, sort, and route the transactions in the file for processing through the appropriate clearing channel. This service is integrated with HSBC's global payments systems, enabling straight-through processing and automated reconciliation reports for clients.
- **J.P. Morgan Access Host-to-Host:** J.P. Morgan's corporate banking portal, Access, includes a Managed File Transfer Service for host-to-host integration. Clients can exchange files with J.P. Morgan via SFTP, HTTPS, or AS2, according to the bank's supported protocols (SSH key or certificate authentication is used). In practice, a company might send an encrypted payment file each morning to J.P. Morgan and receive prior-day account statements (for all their accounts globally) each night in a consolidated file. J.P. Morgan's host-to-host channel supports a wide array of formats, from ACH files and EDI payments to SWIFT MT and ISO 20022 XML messages. Notably, J.P. Morgan has indicated that clients using its file channels (Host-to-Host or Access portal uploads) did not need to immediately migrate their formats during the industry's recent ISO 20022 transition – the bank can internally map legacy formats to new standards. This underscores how banks often shield corporate file users from interbank format changes by doing conversions behind the scenes.
- **Other Examples:** Virtually all major banks and many regional banks have similar offerings. Bank of America's CashPro, for instance, allows file uploads/downloads via SFTP and other methods, and supports formats like BAI2, CSV, and ISO XML.

In Europe, many banks offer both direct SFTP and EBICS connections for clients (where EBICS is prevalent, e.g. Deutsche Bank, BNP Paribas, etc., support EBICS alongside SFTP). For multinational corporates, the SWIFT SCORE service is another avenue: the company can connect through a SWIFT service bureau and send payment files to any of its banks via the SWIFT network (either as individual SWIFT MT messages or bulk FileAct files). This avoids maintaining separate connections to each bank, but comes with additional SWIFT membership overhead. Nonetheless, the common thread is that file-based host-to-host integration is an industry-standard practice – allowing corporate treasury systems to “talk” to bank systems directly. It is highly reliable and scalable, capable of moving large volumes of payments and data daily with minimal errors.

Security and Encryption Standards for File Transfers

Security is paramount when transmitting sensitive financial data like payment instructions or account statements. Secure file transfer protocols incorporate multiple layers of protection:

- **Encrypted Transport Channels:** SFTP (which runs over SSH) encrypts all data in transit, preventing eavesdropping. Similarly, FTPS uses SSL/TLS encryption for the channel, and HTTPS or other WebDAV/API file endpoints also ride on TLS. These protocols ensure that the bulk data files cannot be read or tampered with during transmission. By contrast, traditional FTP was unencrypted and is generally disallowed for modern banking data transfers. Banks will typically only enable encrypted protocols (and often disable older ciphers to meet standards). For example, an SFTP host-to-host connection might require the client to use strong SSH ciphers and MAC algorithms that the bank supports (banks often publish supported cipher lists for compliance).
- **Authentication and Access Control:** Rather than simple passwords, host-to-host connections usually use SSH key authentication or digital certificates to verify the client’s identity. The corporate generates a key pair; the bank’s server keeps the public key to grant access, ensuring only the authorized party (holding the private key) can connect. Some banks also implement IP allow-listing or VPN tunnels so that only connections from the corporate’s known servers can even reach the SFTP service. These measures mitigate the risk of unauthorized access. In addition, many banks support dual-factor authentication for any interactive access or portal-based file transfer, though for automated system-to-system transfers the key/certificate usually suffices as a factor. Role-based access control on the bank’s side further ensures a client can only reach their own directory and files.
- **PGP File Encryption:** In many host-to-host setups, files are encrypted at the file level using PGP (Pretty Good Privacy) in addition to the transport encryption. This means the bulk payment file is encrypted and/or digitally signed by the sender before transmission, and the bank decrypts it upon receipt (having exchanged PGP public keys beforehand). PGP encryption provides an extra layer of security (protecting the file contents even if, for instance, it were misrouted or stored on an

intermediary system) and assures the file's integrity and origin through digital signatures. Citi, for example, explicitly requires or encourages PGP encryption for files sent via CitiConnect, on top of SFTP. Industry-wide, it's common that "the connexion is generally based on SSH keys...and files are usually signed and encrypted using PGP keys" over host-to-host channels.

- **Digital Signatures & Non-Repudiation:** Beyond PGP, certain networks demand formal digital signatures on messages. EBICS, for instance, uses X.509 certificates to sign each file and may require multiple signatures (from authorized users) for sensitive transactions. Similarly, corporates using SWIFTs FileAct or FIN may employ hardware token-based signatures (e.g. SWIFT 3SKey) to sign payment files/messages. These signatures ensure that the file was sent by an authorized source and has not been altered, providing non-repudiation. In cases where a file is signed but not yet approved, some protocols (like EBICS T) will hold it until a user approves it through a separate channel. However, most automated SFTP flows are set up to allow straight-through processing once the secure file is received.
- **Integrity and Audit Trails:** Banks often generate acknowledgments or hash totals to confirm file integrity. For example, when a file is received, the bank might return a confirmation file or message (sometimes a control report or a "ACK/NACK" file) verifying receipt and whether the file passed validation. This is part of the overall control framework to ensure nothing is lost or duplicated. All file transfers are logged extensively by banks for audit purposes. Secure file transfer systems keep an audit trail of who sent what and when, which is crucial for compliance audits and troubleshooting.

Overall, these security practices make SFTP and related file transfers very robust. As one industry source notes, encryption, authentication, and access controls are mandatory – secure file transfers "protect sensitive data with encryption... while avoiding fines and audit failures" under strict data regulations. The combination of transport security (SSH/TLS), file encryption (PGP), strong authentication (keys/certs), and integrity checks ensures that file-based payment transmissions meet the high confidentiality and integrity requirements of the finance industry.

Regulatory and Compliance Considerations

Using FTP/SFTP and file-based flows in banking must align with various regulatory and compliance standards:

- **Data Protection (GDPR):** Banks in jurisdictions like the EU must handle personal and financial data in accordance with privacy laws such as the GDPR. This means that customer data contained in payment files or reports must be protected during transit and storage. The secure file transfer mechanisms described (encryption in transit, PGP encryption at rest, strict access controls) are designed to ensure compliance by keeping personal data confidential. Without such measures, a file containing, say, salary payments could expose sensitive personal information. GDPR also affects cross-border data transfers – banks need to ensure that file exchanges across borders have proper safeguards or legal bases. Secure file transfer solutions often come with built-in compliance features (encryption standards, audit logs) to help meet these requirements. In short, using SFTP with strong encryption and authentication is a key way to avoid data breach risks that could lead to regulatory fines under GDPR or similar laws.
- **Payment Services Directive 2 (PSD2) and Open Banking:** PSD2 (in effect since 2018–2019 in Europe) aimed to open up bank data and payment initiation to third-party providers via APIs. While PSD2 primarily concerns API access (not file transfers), its philosophy of secure customer authentication and data sharing influences bank connectivity. Banks have developed APIs for account information and payment initiation to comply with PSD2's mandate. However, corporate host-to-host file transfers were generally outside the PSD2 scope since they involve the client's own data, not a third-party. Many large corporates continue using SFTP/host-to-host channels for bulk payments despite PSD2, because those channels can be more feature-rich and do not require interactive authentication for each transaction (unlike PSD2 APIs that often require strong customer auth for each payment). That said, PSD2 has spurred banks to also offer premium corporate APIs (often using REST/JSON with OAuth2 and mTLS security) as modern alternatives alongside SFTP. Open Banking APIs might be used for real-time needs or fintech integrations, but for bulk, recurring corporate flows, SFTP remains prevalent due to its reliability and the fact that treasurers often only need intraday or daily data rather than true real-time updates. Banks must ensure that whichever channel – API or file – they provide meets security standards equivalent to regulatory expectations (for instance, ensuring an SFTP key-based login is as secure as a PSD2 API token in practice).
- **ISO 20022 Migration:** The banking world is in the midst of migrating payment messaging from legacy formats (like SWIFT MT) to the ISO 20022 XML standard. For corporate-to-bank file transfers, this means banks need to support ISO 20022 file inputs and outputs. Many banks have been "ISO 20022 ready" for years in the context of SEPA and instant payment schemes, and now for cross-border payments. Corporate clients are encouraged to use the new formats (which can carry richer data), but importantly, banks often continue to accept legacy file formats and will do conversions internally to comply with clearing network requirements. For example, as SWIFTs cross-border network moved to ISO 20022 in 2023, a corporate that was sending an MT101 file via SFTP could still do so – the

bank would map it to an ISO 20022 message for actual transmission, meaning the corporate was exempted from a forced migration at that time. Nonetheless, new implementations increasingly use ISO 20022 XML for consistency. Banks like Citi report that the majority of new file flows use ISO XML formats. Ensuring that file transfer channels can handle larger message sizes and new data elements (like longer references, richer remittance info) has been part of ISO 20022 readiness. Compliance with ISO standards also feeds into future-proofing these channels for upcoming regulations (like alignment with ISO-based reporting for sanctions screening, etc.).

- **Local Regulations and Standards:** Different countries may impose specific security or format requirements on electronic banking. In Germany and France, for instance, EBICS is mandated or strongly encouraged for corporate bank connectivity in certain contexts. In the US, banks must follow FFIEC guidance on authentication for high-risk transactions – using SFTP with keys could be seen as meeting multifactor auth requirements for electronic payments. Additionally, sectoral regulations (like SOX for audit trails, or PCI-DSS for protecting card data in files) can come into play. Banks and corporates must ensure that file content (e.g., if files contain card numbers or personal data) is handled in compliance with these rules. The importance of secure file transfer for compliance is underscored by the fact that non-compliance (e.g., sending unencrypted sensitive data) can lead to heavy penalties or audit failures. Thus, managed file transfer systems are typically audited and certified to meet standards, providing encryption (AES-256, etc.), integrity checks, and audit logs as required by various laws.

In summary, while using FTP/SFTP is a technical choice, it is deeply entwined with compliance. Secure file channels, properly implemented, allow banks and corporates to meet regulatory mandates on data security and to smoothly adopt new payment standards like ISO 20022. They provide a controlled, auditable environment for moving financial data.

The Shift to APIs and Real-Time Services: Is SFTP Being Phased Out?

With the rise of real-time payments and open banking, many wonder if file-based transfers are becoming obsolete. The reality is a gradual evolution rather than an overnight replacement. APIs and instant connectivity are indeed expanding, but FTP/SFTP-based methods continue to have important roles:

- **APIs vs. File Transfers:** APIs enable transaction-by-transaction interactions and immediate feedback. For example, an API can let a corporate system trigger a single payment and receive a real-time confirmation or fetch an up-to-the-minute balance on demand. This level of immediacy is valuable for certain use cases (e.g. e-commerce payments, just-in-time liquidity checks). In contrast, SFTP file transfers operate on a batch paradigm – they excel at moving large volumes of data

in one go, but typically on a schedule (e.g. end-of-day or hourly) or triggered by a file drop event. Treasury professionals note that for many purposes, daily or intraday batch data is sufficient; having up-to-the-second data via API might not yield substantial practical benefit for activities like daily cash positioning. Thus, both methods can coexist, each suited to different needs. Crucially, APIs require both the bank and corporate systems to support the integration; not all banks have full-featured corporate APIs yet, and not all corporate IT environments are ready to consume them. SFTP, being an older and very stable technology, is universally supported and understood.

- **Regional Adoption:** In some regions, we see a faster shift. For instance, European banks (spurred by PSD2) have developed open banking APIs for single payments and account info accessible to third-party providers. Additionally, initiatives like EBICS 3.0 aim to modernize file transfer (standardizing it across countries) but still operate in batch mode. Meanwhile, newer real-time payment systems (UK Faster Payments, US RTP, etc.) often provide or even require API-based submission to achieve instant processing. Banks respond by offering new connectivity: some have built premium APIs for corporate clients, which go beyond the basic PSD2 requirements, allowing corporates to initiate payments or retrieve reports programmatically in real time. These typically use REST/JSON over HTTPS with strong OAuth2/mutual TLS security. For example, a bank might offer an “instant payments API” for initiating a single immediate credit transfer, which a treasury system could call in urgent cases instead of waiting for the next file batch.
- **Hybrid models:** Most enterprises are adopting a hybrid connectivity model. They might use SFTP file transfers for high-volume routine processes (e.g. payroll files, daily sweeping of account balances) where the batch approach is efficient and reliable. Simultaneously, they might use APIs for specific needs like retrieving intraday transaction updates or sending low-volume, high-urgency payments. As one 2025 treasury technology commentary notes, secure file transfer remains the “all-weather road” that reaches every bank, whereas APIs are like high-speed lanes that exist in some places but not universally. Many large bank portals (J.P. Morgan Access, Bank of America CashPro, etc.) still have limited API capabilities, so file transfer is the fallback that guarantees coverage of all accounts and transaction types. A hybrid strategy ensures no gaps in connectivity – if an API isn’t available or fails, the file channel covers it, and vice versa.
- **Modernization, Not Elimination:** Banks are modernizing their file transfer platforms rather than eliminating them. For example, some banks are moving from older connectivity like leased-line FTP or antiquated EDI networks to internet-based SFTP with better encryption, or from proprietary text formats to XML, but the fundamental concept of file exchange persists. Additionally, managed file transfer (MFT) solutions are being upgraded to integrate with cloud services and to provide event-driven file processing (so files can be processed as soon as they arrive, narrowing the gap between batch and real-time). Meanwhile, interbank file networks are also accelerating – e.g. same-day ACH batches now process multiple times per day, and SEPA files clear in cycles throughout the day, reducing wait times.
- **EBICS and Other Protocols:** Protocols like EBICS can be seen as bridging the gap – they offer a standardized, secure file exchange (with acknowledgement flows) that’s more uniform than custom SFTP setups, but still essentially batch. Open

banking and ISO 20022-based APIs are partly intended to supersede regional protocols like EBICS in the long run, yet in 2025 EBICS remains heavily used in its core markets. It's telling that even as EBICS 3.0 was introduced (to harmonize the standard across Europe), banks have continued to invest in it, indicating file-based transfers will be around for the foreseeable future.

In conclusion, while APIs and real-time services are growing rapidly and offer advantages for immediacy, SFTP and secure file transfers are not being wholesale phased out. They continue to carry the bulk of scheduled corporate payment traffic and reporting data due to their reliability, capacity, and established trust. The future likely lies in complementary use: critical transactions and data queries may go through APIs for instant processing, whereas high-volume, less time-sensitive exchanges remain file-based. Treasury teams value the resilience of file channels – indeed, many fintech connectivity providers advertise a blend of “APIs where available, files for universal coverage” to ensure every bank can be reached. Banks likewise maintain their host-to-host offerings alongside newer API portals to cater to all client preferences. Given ongoing innovations, we may see a gradual convergence (for example, API calls that send batch files, or files that trigger near-real-time processing), but in 2025 the secure file transfer methods (SFTP, EBICS, etc.) firmly retain their place in the financial ecosystem, underpinning mission-critical fund transfers and reporting processes.