



APPX LIBRARY

APPX Data File Encryption

APPX 6.2.1 | All supported platforms

Encrypt and decrypt APPX datafiles. File and database exercises, CSV export, analysis, recovery, and key management. At publication on October 16, 2026, APPX data file encryption has no known bugs in release 6.2.1. Earlier releases are known to have bugs.

Contents

Chapter 1 Before starting	3
What this manual covers	3
What can be encrypted	4
Encryption technology	4
Passphrase, encryption keys, and index files	5
Keep the keys safe	6
Chapter 2 Prepare the test system	7
Back up and record the starting state	7
Set the system passphrase	7
Open DMO Data File Management	9
Chapter 3 Encrypt PROSPECT	10
Select and export only PROSPECT	10
Set the file key and replace the file	11
Export the encrypted file to CSV	12
Chapter 4 Decrypt PROSPECT	14
Export before clearing the file key	14
Replace and verify the decrypted file	15
Chapter 5 Encrypt the DMO database	16
Inventory and export the eligible files	16
Create and assign a dedicated encryption group	17
Replace and check every converted file	19
Chapter 6 Decrypt the DMO database	21
Export every affected file first	21
Clear the encryption settings and replace	21
Verify the decrypted database	22
Chapter 7 Analysis and recovery	24
Check an encrypted file through APPX	24
Analyze a working copy	25
Decide whether to rebuild a missing or damaged index	26
Physical recovery	27
Extract records flagged as deleted	28
Test recovery on a disposable copy	29
Chapter 8 Routine operation and compatibility	30
Backups, restoration, and another server	30
Share encrypted files across servers	30
Change a key or the system passphrase	31
APPX ODBC and external database interfaces	31
Check ODBC access before converting a file	32
ODBC access in the 6.2.1 test	33
Unicode in APPX 6.2.1	34
When a check fails	35
Appendix A Reference and completion checks	36
Desktop Client keys	36
Conversion completion record	36
Further reading	36

CHAPTER 1

Before starting

Understand the scope and protect the keys before making a change.

What this manual covers

APPX data file encryption protects data stored in native APPXIO files. This manual covers encrypting and decrypting an existing file, encrypting and decrypting the eligible files in a database, exporting encrypted data to CSV, and checking and recovering encrypted files.

This manual covers APPX 6.2.1 on any supported platform. The encryption and decryption workflow is the same across platforms. Command examples are provided for Unix and Windows where needed. GUI illustrations are retained from the earlier edition where the controls and steps are unchanged; screen appearance and sample data may differ.

At the time of publication, October 16, 2026, APPX data file encryption has no known bugs in release 6.2.1. Earlier releases are known to have bugs. Consult the APPX bug tracker for details.

Use a separate test installation and a copy of the DMO sample database. Complete each exercise and check its result before continuing. Keep application users, scheduled jobs, API users, and ODBC users out of the test database during a conversion.

Before converting a file used by ODBC clients, complete [Check ODBC access before converting a file](#). Native APPX access and ODBC access must be checked separately.

Exercise	Starting state	Result
Encrypt PROSPECT	PROSPECT is unencrypted	Only PROSPECT is encrypted
Decrypt PROSPECT	PROSPECT is encrypted with its own file key	PROSPECT is unencrypted again
Encrypt DMO	Eligible DMO files are unencrypted	The eligible files use a dedicated encryption group
Decrypt DMO	The dedicated group encrypts the eligible files	Those files are unencrypted again

The CSV and recovery exercises can be run while PROSPECT is encrypted. The remaining chapters explain routine operation, backups, key changes, Unicode, and APPX ODBC.

What can be encrypted

Use this feature for native APPXIO indexed files and variable-length consecutive files. One-record files and fixed-length consecutive files cannot be encrypted with this feature.

In the standard DMO application, the nine permanent indexed files used here are ACTIVITY, ACTTYPE, CONTACT, COUNTRY, INDUSTRY, OPPORTUN, PROSPECT, SALESREP, and STAGE. PARAM is a one-record file. STREAM is a disk-subprocess stream definition. Leave PARAM and STREAM out of these conversion selections.

“Encrypt DMO” therefore means encrypt every eligible native datafile in DMO. It does not mean that the entire database directory, every file organization, or every generated output becomes encrypted.

An APPX database can contain files from more than one application. Inventory the database's related applications before starting. Repeat the database exercise for every application with eligible files in DMO. The nine-file selection shown here covers the standard DMO application; additional applications and locally added files require their own selections and checks.

APPX data file encryption is not intended for data held in an external datastore and will not encrypt that data. This includes Oracle and SQL Server tables accessed through an APPX external database interface. Configure encryption for those tables and their backups in the database product that stores them.

Encryption technology

APPXIO data file encryption uses AES-256 by default, with a **256-bit (32-byte) key**. AES is an established symmetric encryption standard published by NIST. That cryptographic key size is distinct from the number of characters entered in an APPX passphrase or file-key field. See [NIST's Advanced Encryption Standard](#).

Encryption and decryption run **inside the APPX executable process**, as part of native APPXIO file access. APPX encrypts records when writing them and decrypts them when reading them. This integration protects stored native datafiles while ordinary APPX applications continue to use normal record values. Protect the system passphrase and file or group keys to retain control of access and restoration.

Passphrase, encryption keys, and index files

Three different things need to be kept straight.

Item	Purpose	Where it is managed
System Encryption Passphrase	Enables encryption and is part of the configuration needed to access encrypted files	System Administration, System Setup, System Parameters
File or FMS group Encryption Key	Selects encryption for the file or group	File Attributes or FMS Group Attributes
A file named PROSPECT.key	Holds the APPX indexes for PROSPECT.dat	Maintained by APPX file utilities

The `.key` file is an index file. It is not a saved copy of the Encryption Key entered on the screen.

Setting a passphrase or entering a key does not rewrite an existing file. An existing file is converted by exporting its records under the old settings and importing them with **Replace** under the new settings. A Merge import does not perform that conversion.

File Attributes shows the configuration, rather than proving the physical file has been converted. To confirm stored encryption, use DUMP HEADER on a complete working copy as described in [Analyze a working copy](#). The header must show ENCRYPTED after encryption, and that flag must be absent after decryption.

A nonblank file-level encryption key takes precedence over a group key. A blank file-level key allows the group key to apply. Clearing a file key will not decrypt a file that still inherits a nonblank group key.

Keep the keys safe

Save the exact system passphrase and every file and group key before using them. Include the APPX system, customer registration, database, application, file or group name, effective date, and key version in the record. Treat capitalization, punctuation, and every character as significant.

An organization-managed password manager such as 1Password Business is a reasonable place for these records. Use a restricted vault, individual accounts, and multifactor authentication. Give access to the people responsible for restoration, with enough coverage that one person's absence does not prevent recovery. See [1Password's vault access instructions](#).

Keep an independent recovery route for the password manager itself. Store the appropriate Emergency Kit or other approved recovery material securely outside the APPX server, and arrange account recovery before it is needed. Test that the designated administrators can regain access. See [1Password's organization recovery plan](#).

Use generated, unpredictable values that fit the APPX entry fields. Generated printable ASCII values are convenient when keys must be entered on systems with different keyboard layouts. APPX's Random Key button is also available. Save its result in the vault before leaving the screen.

Keep key records out of email, support tickets, screenshots, shell commands, scripts, and ordinary document folders. Retain old key versions for as long as a backup may require them. Keep a protected backup of System Administration as well as the application data. A backup of only the encrypted datafiles is not the complete restoration kit.

Do not change an existing system passphrase to try an exercise. It applies to the APPX system, including encrypted files in other databases. Use an isolated test system or retain the system's existing passphrase and follow its established key procedures.

CHAPTER 2

Prepare the test system

Back up the test system, set the passphrase, and open DMO file management.

Back up and record the starting state

1. Confirm that this is a separate test installation and that no production users or services use its data or System Administration files.
2. Back up the test database and its matching structures. For indexed files, preserve both `.dat` and `.key` files.
3. Back up the System Administration files and retain the applicable registration and key records securely.
4. Record each file's organization, effective FMS type, FMS group, optional path, and any file-level key setting. Check inherited settings as well as values shown directly on the file.
5. Run Verify Files Menu, Verify Integrity of APPXIO Files. Resolve an existing integrity failure before attempting a conversion.
6. Record the export counts and a few recognizable sample records. Use application totals where available.

The sample data shown in the GUI illustrations contains 7 PROSPECT records and 125 records across the nine eligible DMO files. Your counts may differ. Compare your before and after counts, rather than changing your data to match the examples.

Paths, operating system accounts, APPX user IDs, registration details, and service ports vary by installation. Use your own values. Command examples use `/opt/appx` on Unix and `C:\APPX` on Windows; neither is a required installation location.

Check `APPXPATH`, any `APPX_OSA_PATH` override, and FMS path overrides when locating backups or exported files. An overridden file may be stored outside the usual directory tree.

Set the system passphrase

Skip setting a new passphrase if the test system already has the correct one. Keep its existing value and confirm that its recovery record is available.

1. Open System Administration, System Setup, System Parameters.
2. Switch to **Change** mode. The Change button beside Encryption Passphrase cannot be used in Inquire mode.
3. Click **Change** beside Encryption Passphrase.

System Parameters ? ✓ ✕

Site Name

Direct Process 1-

Database ID

Application ID

Process

FMS Group

Date Mask 06/05/0807 04:03 02.01

Currency Symbol \$ Position 1 (1=Leading, 2=Trailing)

Decimal Notation . Grouping ,

Language ID

Auto Select All Files? ☒ Design Xfer Retain Audit Default? ☒

Minimize Term I/O? ☐ Allow Design Default? ☒

Create DF Files? ☐ SCCS Active? ☐

Encryption Passphrase (no pass phrase set)

Figure 1. System Parameters. Set the passphrase only on the isolated test system.

4. Enter the new passphrase in both fields on Set/Change Pass Phrase.
5. Accept the dialog and save System Parameters. If the report defaults dialog appears, accept the appropriate existing defaults to finish the process.
6. Keep the exact passphrase in the restricted key record.



Figure 2. Set/Change Pass Phrase. The encryption inventory report is available here before a planned passphrase change.

The passphrase enables the feature. No existing datafile has been encrypted by this step.

Open DMO Data File Management

The following route provides the database and application context used in the screenshots.

1. From the APPX main menu, choose Design an Application.
2. Select application DMO, version 00. If the chooser contains many applications, enter DMO in its application filter, use Set Filter, and open DMO.
3. Open the Tools tab and choose File Management under Physical File Management.
4. Enter database DMO and accept the database entry. Confirm that Application is DMO and that its description is Demo Application.
5. Open File Selection before running a utility. Check the selected-file count each time you return to Data File Management.

These exercises use database DMO, application DMO, version 00. Use the appropriate identifiers for your own database and application. Do not confuse the application version with the database ID.

CHAPTER 3

Encrypt PROSPECT

Encrypt one file and export it while it remains encrypted.

Select and export only PROSPECT

1. Open DMO Data File Management and File Selection.
2. Choose Unselect All.
3. Use Go To to locate database DMO, application DMO, version 00, file PROSPECT. Accept the locator.
4. Use SELECT to toggle PROSPECT into the selection. In this Desktop Client, use F3. Return to Data File Management with END.
5. Confirm that exactly **1 of 11** files is selected in the standard sample. Check that the selected file is PROSPECT. Your total file count may differ.
6. Choose Export Files (CSV). Leave Include Headings unchecked for the conversion copy. Click Proceed.

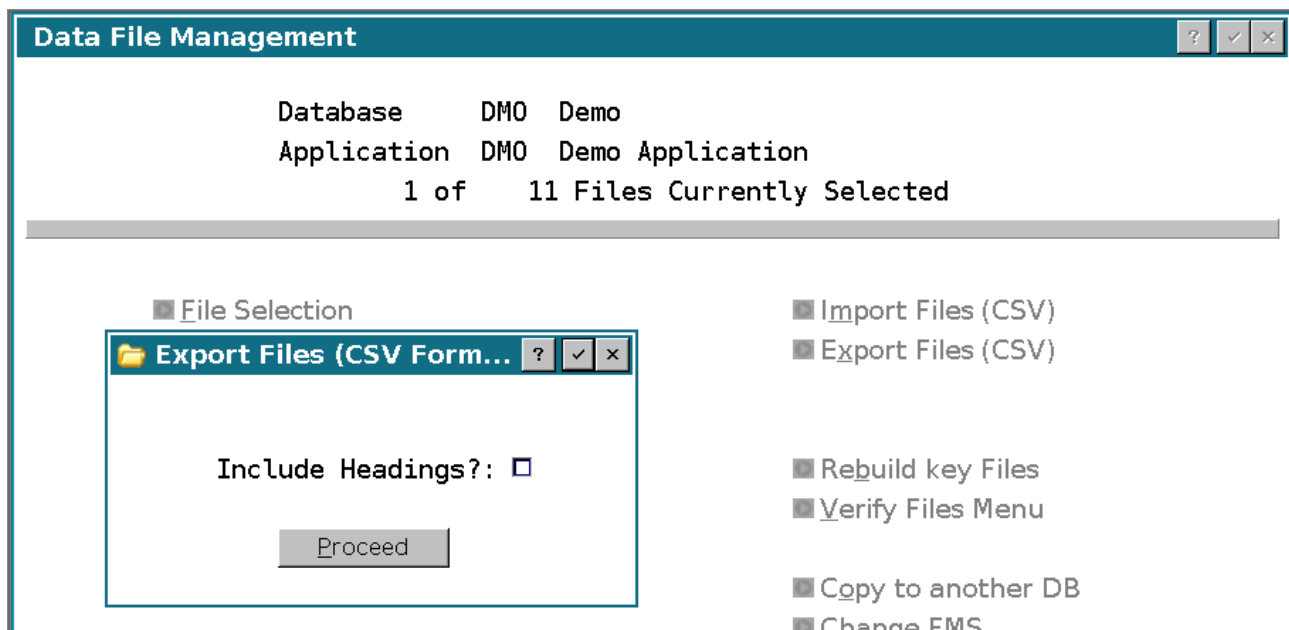


Figure 3. Export Files (CSV) with one file selected. Keep the conversion copy without headings.

7. Confirm that the log identifies PROSPECT and reports every record exported. The example reports Exported 7 of 7 records.
8. Preserve this CSV in a restricted location. Record its full pathname and the export count before changing any encryption setting.

The normal export location in this test is `APPXPATH/DMO/DMO/Portdata/PROSPECT.csv`. Your installation's root and any configured export path may differ. Verify the actual output location and keep a protected copy outside the working export folder, since another export can replace the working CSV.

Set the file key and replace the file

Create PROSPECT's file key here. The [earlier setup](#) established the system passphrase; it did not create a file key. Use a separate generated value for PROSPECT.

1. Return to Data File Management and choose File Specifications. Keep only PROSPECT selected.
2. Switch to Change mode if necessary. Record the existing FMS type, group, and optional path. The independent PROSPECT exercise uses no encryption group.
3. Choose File Attributes. For this exercise, click **Random Key** to have APPX generate a new value in **Encryption Key**. Save that exact value in the restricted vault described in [Keep the keys safe](#), identifying database DMO, application DMO, version 00, and file PROSPECT. Confirm that the saved value matches the field before accepting the screen.

If the site's key procedure requires a password-manager-generated key instead, generate a new value in that manager, save it with the same file identification, and enter that exact value in Encryption Key. In that case, leave Random Key alone: clicking it would replace the value just entered. Do not enter the system passphrase in this field. Both choices supply the file key; they do not select different encryption methods.

Figure 4. PROSPECT File Attributes before a key is entered. Save the actual key securely; keep it out of screenshots.

4. Accept File Attributes and save the PROSPECT specification. Do not change the FMS type or optional path as part of enabling encryption.
5. Return to Data File Management. Confirm the one-file selection again.
6. Choose Import Files (CSV). Set Disposition of Native File to 2 = **Replace** and click Proceed.

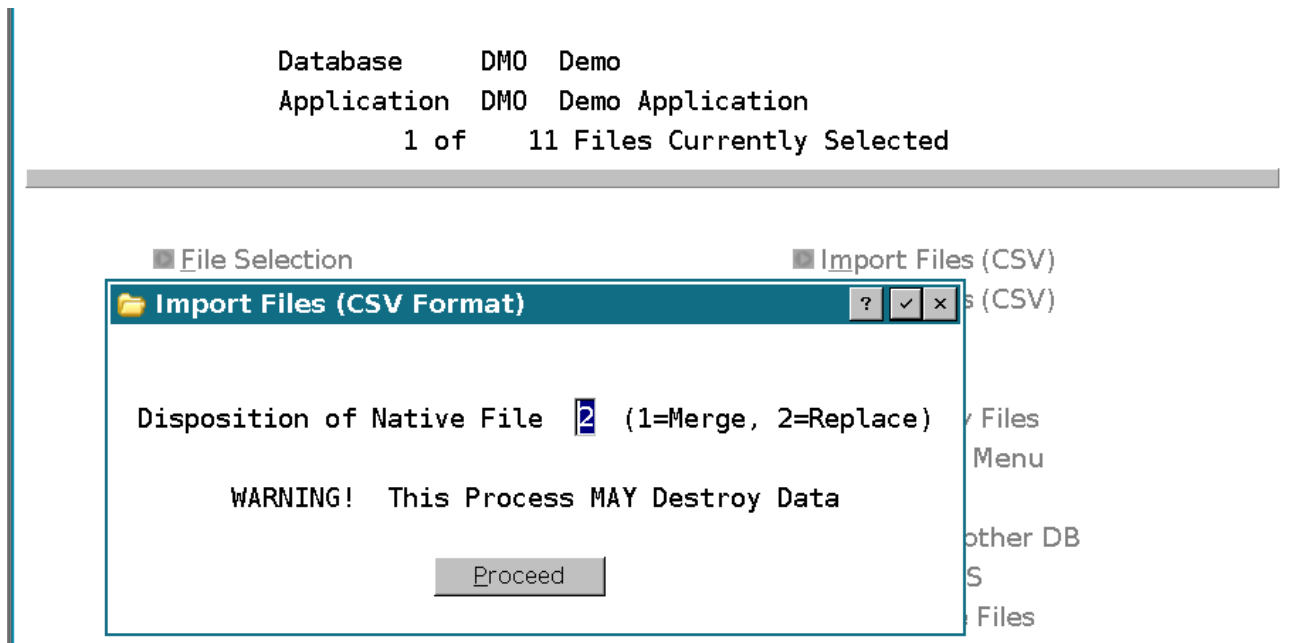


Figure 5. Import Files (CSV). Replace recreates the file under the current encryption settings.

7. Confirm that every exported PROSPECT record was imported and that no error appears in the log.
8. Run Verify Files Menu, Verify Integrity of APPXIO Files. Compare the record count with the export count.
9. Open PROSPECT in the DMO application and compare the sample records recorded before conversion.

The File Specifications **Records** column is not the result count to use for this check. Compare the export, import, and verification results.

In this exercise, APPX imported all 7 records and the recreated physical datafile was confirmed encrypted. Other DMO files were left under their previous settings.

Export the encrypted file to CSV

Yes, an encrypted APPXIO datafile can be exported to CSV. APPX reads it through the normal file manager and writes the record values to the export. There is no need to decrypt the stored file first.

1. Keep PROSPECT encrypted and correctly configured.
2. Select only PROSPECT in Data File Management.
3. Choose Export Files (CSV), retain the same heading choice used for the original comparison copy, and click Proceed.
4. Check the exported-record count and compare the CSV with the original export. For a spreadsheet copy, a separate export can include headings.

Export Files (CSV)				
DB	Ap	V...	File	Status/Message
DM0	DM0	00	PROSPECT	Exported 7 of 7 records

Figure 6. Exporting encrypted PROSPECT through APPX. All 7 records were exported.

The headerless CSV exported while PROSPECT was encrypted matched the original export byte for byte in this test. Exporting did not remove encryption from the stored APPXIO file.

The CSV itself is unencrypted. Restrict its directory and any copies, use an approved protected transfer method, and remove temporary copies according to the site's retention procedures after verification. Include spreadsheet copies, downloads, shared folders, and backups of export folders in that handling. Keep the untouched conversion export separate from a copy edited in a spreadsheet.

CHAPTER 4

Decrypt PROSPECT

Recreate PROSPECT without encryption and check its records.

Export before clearing the file key

This exercise reverses the independent file-key example in Chapter 3. It starts with PROSPECT encrypted, no inherited group key, and a working system passphrase.

1. Stop users and jobs that can access PROSPECT.
2. Back up the encrypted file pair, matching structure, and current administration configuration.
3. Select only PROSPECT and export it with Export Files (CSV), without headings.
4. Check the export count and preserve the CSV before changing the key.
5. Open File Specifications, File Attributes in Change mode and clear Encryption Key.



Figure 7. PROSPECT's file key cleared after export. The physical file is still encrypted until the replacement import.

6. Accept File Attributes and save the file specification.

Leave the system passphrase in place. Decrypting one file is not a reason to change the passphrase used by the rest of the system.

If PROSPECT instead inherits a group key, clearing its file-level key is insufficient. After exporting it, move it to an equivalent group without encryption, preserving the effective FMS, path, and other required attributes. Clear a group's key only when every affected member is included in the planned conversion. Chapter 6 uses a dedicated group for that purpose.

Replace and verify the decrypted file

1. Return to Data File Management and confirm that only PROSPECT is selected.
2. Choose Import Files (CSV), set 2 = **Replace**, and click Proceed.
3. Confirm that every exported record was imported.
4. Run Verify Files Menu, Verify Integrity of APPXIO Files.
5. Reopen the sample records in DMO and compare them with the pre-conversion values.
6. Make another headerless CSV export and compare it with the saved export.

PROSPECT was recreated without the encrypted-file flag in this test, and all 7 records were retained. The later database round-trip comparison also confirmed the original PROSPECT CSV values.

Keep the old key and passphrase record while encrypted backups remain in retention. Decrypting the current file does not decrypt an older backup.

CHAPTER 5

Encrypt the DMO database

Use a dedicated group to encrypt every eligible DMO datafile.

Inventory and export the eligible files

Start with the eligible DMO files unencrypted. Complete the PROSPECT decryption exercise first if following this manual in sequence.

1. Inventory every application related to database DMO. Include each application's eligible permanent APPXIO files in the conversion plan.
2. Back up the database, matching structures, and System Administration. Record current FMS settings and any individual encryption keys.
3. In DMO Data File Management, open File Selection and choose Select All.
4. Use SELECT to exclude PARAM and STREAM. Check the organization and file type shown on each row before excluding it.
5. Return to Data File Management. The standard DMO application should have 9 of 11 files selected.
6. Choose Export Files (CSV), leave Include Headings unchecked, and click Proceed.
7. Check every result row. Preserve all nine CSVs and their counts before assigning encryption settings.

Database DMO Demo

Application DMO Demo Application

9 of 11 Files Currently Selected

File Specifications

DB	Ap	File	Records	FMS	FMS Group	Use DB	--Opts-- FA OP FO
DMO	DMO	ACTIVITY	100	9			
DMO	DMO	ACTTYPE	100	9			
DMO	DMO	CONTACT	100	9			
DMO	DMO	COUNTRY	100	9			
DMO	DMO	INDUSTRY	100	9			
DMO	DMO	OPPORTUN	100	9			
DMO	DMO	PROSPECT	100	9			
DMO	DMO	SALESREP	100	9			
DMO	DMO	STAGE	100	9			

Figure 8. The nine eligible DMO files in File Specifications. PARAM and STREAM are outside this selection.

The sample counts are shown below for comparison with the screenshots. Use your own before and after counts if the sample has been changed.

File	Records in this test
ACTIVITY	33
ACTTYPE	13
CONTACT	16
COUNTRY	7
INDUSTRY	28
OPPORTUN	7
PROSPECT	7
SALESREP	10
STAGE	4
Total	125

Repeat the export for additional applications in DMO before changing any setting that will affect them. Do not infer database-wide coverage from a screen whose Application field still names only DMO.

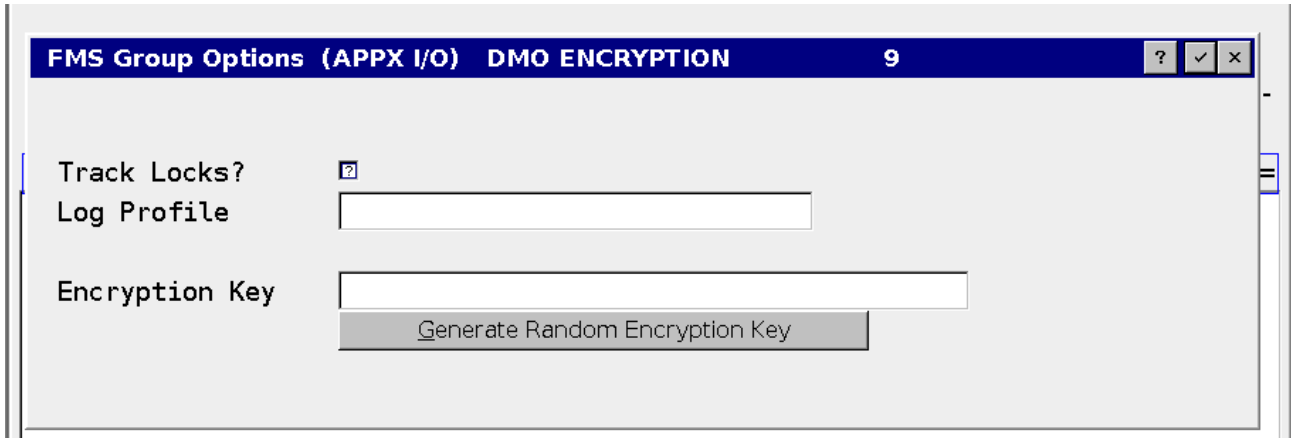
Create and assign a dedicated encryption group

A dedicated group makes the conversion's membership clear. It also allows this database to be decrypted later without changing an unrelated group's key.

For example, a payroll database might keep EMPLOYEE, PAYCHECK, and BANKINFO in a dedicated PAYROLL ENCRYPTION group. Assigning that group lets those eligible APPXIO files share one managed key while preserving the required native FMS and storage path. When the payroll key is changed, export and replace every member together. Keep a separate training database in its own group so a training exercise does not change the payroll files' effective key. A group is a settings convenience; users still need the appropriate application and file permissions.

1. Open System Administration, Configuration, File System Groups under File Maintenance.
2. Add a group named **DMO ENCRYPTION**. This is an example group name; use a distinct name appropriate to your installation.
3. Use the same effective native APPXIO FMS type as the files being converted. The test files use type 9. Your files may use another native APPXIO type. Do not change their storage type merely to enable encryption.
4. Keep the required path and other operational attributes consistent with the original configuration. The test group has no path override.

5. Open FMS Group Attributes. Click **Random Key** to create a new group key, save its exact value in the restricted vault under DMO ENCRYPTION, then accept the attributes and save the group. If the site requires password-manager-generated values, enter a new saved value instead, following the choice explained in [Set the file key and replace the file](#). Use a new key for this group rather than reusing PROSPECT's earlier file key.



The screenshot shows a dialog box titled "FMS Group Options (APPX I/O) DMO ENCRYPTION" with a button labeled "9". The dialog contains three input fields: "Track Locks?" with a checkbox, "Log Profile" with a text box, and "Encryption Key" with a text box. Below the "Encryption Key" field is a button labeled "Generate Random Encryption Key".

Figure 9. A dedicated APPXIO group before its encryption key is entered. FMS type 9 is the test installation's value.

6. Return to the same nine-file selection in DMO Data File Management and open File Specifications in Change mode.
7. Assign DMO ENCRYPTION to each selected file's FMS Group field. Save each changed file specification.
8. Check that no individual file key overrides the intended group key. Any previously encrypted file must already have been exported under its old working configuration before an override is changed.
9. Review the complete list. Every eligible file must have the intended effective group, FMS type, and path.

Database DMO Demo

Application DMO Demo Application

9 of 11 Files Currently Selected

File Specifications

?

✓

✕

DB	Ap	File	Records	FMS	FMS Group	Use DB	--Opts-- FA OP FO
DMO	DMO	ACTIVITY	100	9	DMO ENCRYPTION		
DMO	DMO	ACTTYPE	100	9	DMO ENCRYPTION		
DMO	DMO	CONTACT	100	9	DMO ENCRYPTION		
DMO	DMO	COUNTRY	100	9	DMO ENCRYPTION		
DMO	DMO	INDUSTRY	100	9	DMO ENCRYPTION		
DMO	DMO	OPPORTUN	100	9	DMO ENCRYPTION		
DMO	DMO	PROSPECT	100	9	DMO ENCRYPTION		
DMO	DMO	SALESREP	100	9	DMO ENCRYPTION		
DMO	DMO	STAGE	100	9	DMO ENCRYPTION		

Figure 10. All nine eligible DMO files assigned to the dedicated group. No key value is shown.

This exercise assigns the group directly to the selected file specifications. It does not change a system-wide default. For additional applications in DMO, repeat the assignments for their eligible files. For a file with a different existing path or native FMS requirement, use a suitable dedicated group that preserves that requirement.

Replace and check every converted file

1. Confirm that all selected files have successful exports and that the selection still matches the conversion list.
2. Choose Import Files (CSV), set 2 = **Replace**, and click Proceed.
3. Check every result row. Each file must import all of its exported records.
4. Run Verify Files Menu, Verify Integrity of APPXIO Files for the same selection.
5. Check representative records and application totals in DMO.
6. Export the encrypted files again and compare them with the protected pre-conversion exports.
7. Repeat these checks for every additional application included in the database conversion.

Import Files (CSV)				
DB	Ap	V...	File	Status/Message
DMO	DMO	00	ACTIVITY	Imported 33 of 33 records
DMO	DMO	00	ACTTYPE	Imported 13 of 13 records
DMO	DMO	00	CONTACT	Imported 16 of 16 records
DMO	DMO	00	COUNTRY	Imported 7 of 7 records
DMO	DMO	00	INDUSTRY	Imported 28 of 28 records
DMO	DMO	00	OPPORTUN	Imported 7 of 7 records
DMO	DMO	00	PROSPECT	Imported 7 of 7 records
DMO	DMO	00	SALESREP	Imported 10 of 10 records
DMO	DMO	00	STAGE	Imported 4 of 4 records

Figure 11. The nine-file DMO replacement import. Each file must complete successfully.

All nine indexed files were confirmed encrypted in this test. PARAM remained unencrypted. All nine CSVs exported from the encrypted files matched their originals byte for byte.

Record the completed file inventory and key version with the backup. Assign the appropriate encryption group or file key before creating any new eligible file. An explicit assignment to today's nine files does not automatically add a future file to that group.

CHAPTER 6

Decrypt the DMO database

Export all affected files and recreate them without encryption.

Export every affected file first

This exercise reverses Chapter 5. The DMO ENCRYPTION group must be dedicated to the files in the conversion plan.

1. Stop all application, scheduled, API, and ODBC access to the affected files.
2. Back up the encrypted database and its matching structures, administration configuration, and key records.
3. Inventory every member of each group whose encryption key will be cleared. Include members in other applications or databases if any exist.
4. Select the same eligible DMO files used during encryption. Repeat the selection for any additional related applications.
5. Export all affected encrypted files with Export Files (CSV), without headings.
6. Check every export count and preserve every CSV before changing group or file keys.

Do not clear a shared group's key after exporting only the files visible in one application. A group change affects all its members.

Clear the encryption settings and replace

1. Open System Administration, Configuration, File System Groups.
2. Locate DMO ENCRYPTION and switch to Change mode.
3. Open FMS Group Attributes and clear Encryption Key. Save the attributes and the group.

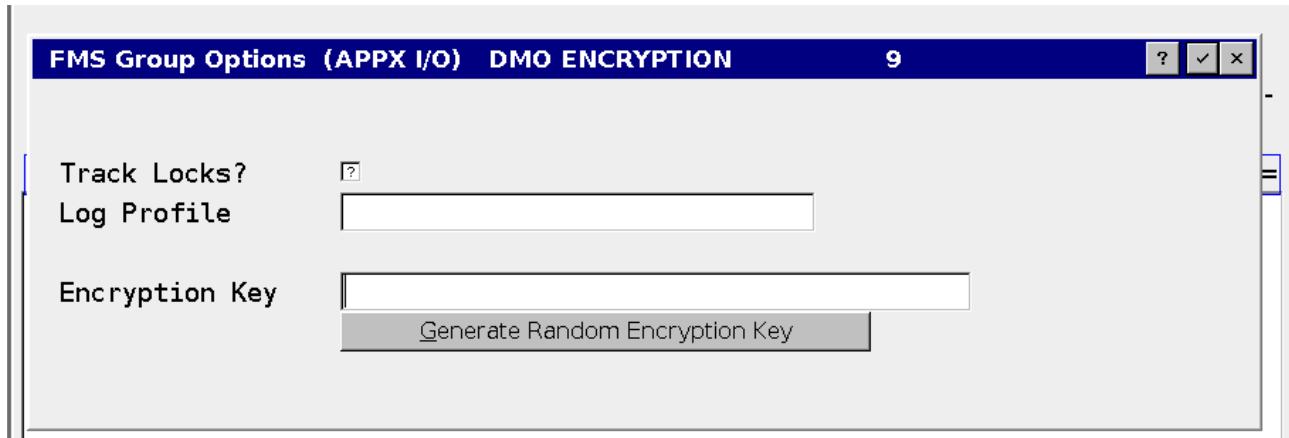


Figure 12. The dedicated group's key cleared after all affected files were exported.

4. Check the affected File Specifications for individual keys. Clear any individual key that would otherwise continue encrypting a file, after confirming its export was made under the old working settings.
5. Preserve the FMS type, path, and other required settings. The group can remain assigned with a blank encryption key.
6. Return to Data File Management and select the same nine eligible DMO files. Confirm the count. Repeat for additional applications in the conversion plan.
7. Choose Import Files (CSV), set 2 = **Replace**, and click Proceed.
8. Check every import count and error message before proceeding to verification.

Leave the system passphrase unchanged. Other databases may still need it.

Verify the decrypted database

1. Run Verify Files Menu, Verify Integrity of APPXIO Files for every converted file.
2. Compare the verified counts with the saved export counts.
3. Check application records and totals.
4. Export the converted files again with the same heading setting and compare them with the saved exports.
5. Confirm that every file on the conversion inventory has been recreated without encryption. Retain the old key records for encrypted backups.

Verify Integrity

DB	Ap	V...	File	Status/Message
DMO	DMO	00	ACTIVITY	OK. Records: 33
DMO	DMO	00	ACTTYPE	OK. Records: 13
DMO	DMO	00	CONTACT	OK. Records: 16
DMO	DMO	00	COUNTRY	OK. Records: 7
DMO	DMO	00	INDUSTRY	OK. Records: 28
DMO	DMO	00	OPPORTUN	OK. Records: 7
DMO	DMO	00	PROSPECT	OK. Records: 7
DMO	DMO	00	SALESREP	OK. Records: 10
DMO	DMO	00	STAGE	OK. Records: 4

Figure 13. Final integrity verification after DMO decryption. The counts match the original sample.

All nine eligible DMO files were confirmed unencrypted after replacement in this test. All 125 records were retained, and each final CSV matched its original byte for byte.

CHAPTER 7

Analysis and recovery

Verify encrypted files and check a recovered working copy.

Check an encrypted file through APPX

Encryption does not remove the need for file-integrity checks. Begin with the GUI utility, using the correct APPX installation, administration configuration, and file selection.

1. Stop changes to the affected files and preserve a backup.
2. Select the files in Data File Management.
3. Open Verify Files Menu and choose Verify Integrity of APPXIO Files.
4. Review every result and record count. Keep the log with the maintenance record.
5. Use Verify Data in Files and application-level checks where appropriate to check values and relationships as well as physical file integrity.

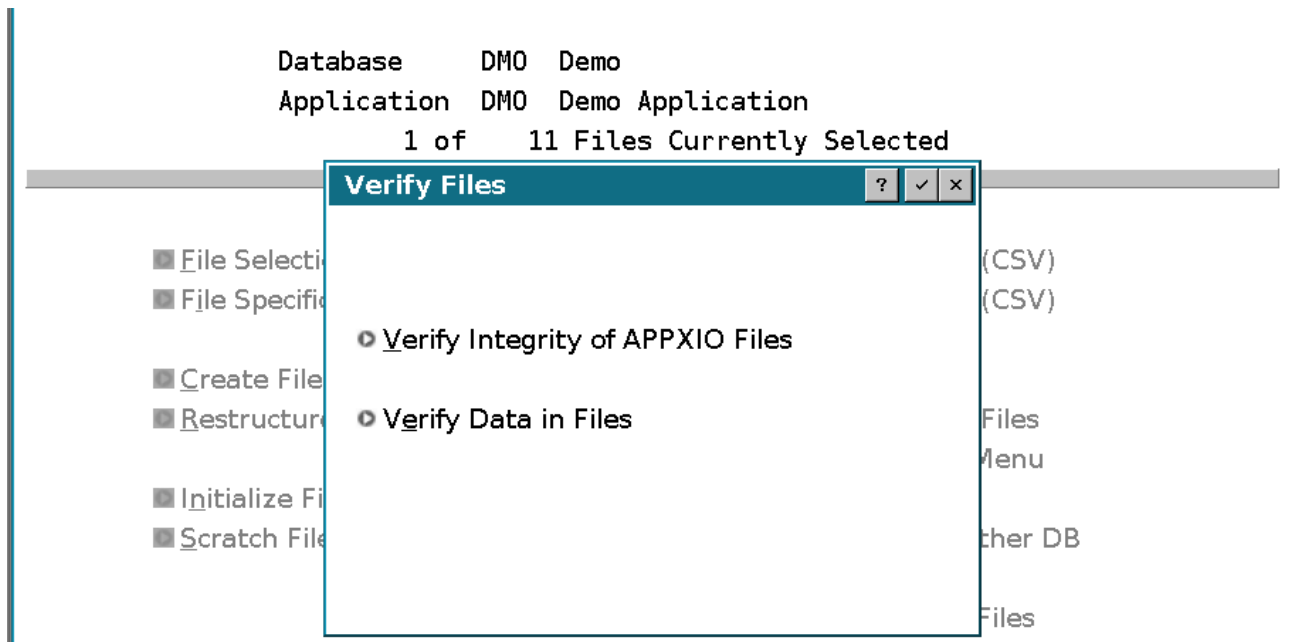


Figure 14. The GUI verification menu. File integrity and data validation are separate checks.

An integrity pass establishes that the structures checked by the utility are consistent. Also compare record counts, recognizable records, and business totals before accepting a conversion or recovery.

If an ordinary file open reports an encryption mismatch, check the system passphrase, effective file or group key, registration, and selected administration tree first. APPX can report FL_CRYPT for an encryption configuration mismatch. Do not generate a new key or replace the file while investigating that condition.

Analyze a working copy

The command-line analyzer is useful for a closer physical check. Its commands are shown as text; the manual's screenshots remain APPX GUI captures.

1. Stop access and make a separate working copy of the complete `.dat` and `.key` pair. Keep the matching structure and original administration context available.
2. Run the analyzer as an account with the required access to that working copy. The utility opens files for read/write access, so preserve the original separately.
3. Use an APPX engine appropriate to the file format, with the originating system's applicable registration available through the correct administration tree.

The following Unix example uses an engine under `/opt/appx`, data under `/opt/appx/data`, and a working copy under `/srv/appx-recovery`. These are example paths and will usually differ from yours. Preserve any required `APPX_0SA_PATH` and FMS overrides.

```
export APPXPATH=/opt/appx/data
/opt/appx/appx -an=/srv/appx-recovery/PROSPECT
```

For Windows PowerShell, substitute your own paths:

```
$env:APPXPATH = 'C:\APPX\data'
& 'C:\APPX\appx.exe' '-an=C:\APPX-Recovery\PROSPECT'
```

At the analyzer prompt, enter these commands exactly, without leading spaces:

```
DUMP HEADER
COUNT REC
VERIFY KEY #ALL
VERIFY FREELIST
QUIT
```

DUMP HEADER identifies the physical file format and its ENCRYPTED flag. COUNT REC reports active and deleted records. VERIFY KEY #ALL checks the indexes against the records, and VERIFY FREELIST checks free-space structures.

Here, an **index-oriented analyzer command** means one that reads the `.key` index structures. **VERIFY KEY #ALL** is the index check used above: it checks all defined indexes against the data records. **VERIFY FREELIST** checks reusable space in the datafile; it is a different check. Use the complete matching pair for the sequence shown here. When the index is missing, follow [Decide whether to rebuild a missing or damaged index](#) first; do not start this analyzer session with an incomplete pair, including for DUMP HEADER.

The encrypted PROSPECT copy in this test showed ENCRYPTED, 7 active records, no deleted records, and 7 entries on each of its two indexes. Treat analyzer output and any record dumps as confidential: authorized utilities can display decrypted values.

Decide whether to rebuild a missing or damaged index

Rebuild key Files reads an existing datafile and constructs its indexes. It does not repair damaged record contents or supply missing encryption settings. Use this sequence to decide whether a rebuild is sufficient.

1. Stop access. Preserve the original `.dat`, any remaining `.key`, matching structure, and administration configuration. Do not pair a current `.dat` with an index from a different backup.
2. Confirm the test database's structure, native FMS, storage path, passphrase, and effective key. Resolve encryption mismatches first; see [Check an encrypted file through APPX](#).
3. Put a copy of the datafile under its expected name in the test database; preserve the damaged index separately. Select only that file in Data File Management and choose **Rebuild key Files**. A missing index alone warrants trying this on a copy before physical recovery.
4. Read the complete rebuild log. Retain any unreadable-record, damaged-record, duplicate-key, or other error. If any appears, follow [Physical recovery](#); do not accept the new index merely because it exists.
5. After a successful rebuild, run **Verify Integrity of APPXIO Files** and [Analyze a working copy](#) on the matched pair. Compare counts and complete exported values with a trusted export or backup. Check recognizable records, business totals, and application relationships.

A successful rebuild and these checks support treating the problem as index damage. If checks fail or required records are missing, keep the result separate and restore a verified backup or follow the recovery procedure. Rebuilding alone does not establish that every expected record is present.

The GUI example below shows an encrypted PROSPECT copy with six active records and one record deleted through APPX. Rebuilding the missing index restored access to the six active records. Rebuilding an index does not restore a deleted record.

Rebuild Key Files				
DB	Ap	V...	File	Status/Message
DM0	DM0	00	PROSPECT	Key File Rebuilt

Figure 15. Rebuild key Files completed for the original encrypted PROSPECT datafile with its index missing.

Verify Integrity				
DB	Ap	V...	File	Status/Message
DM0	DM0	00	PROSPECT	OK. Records: 6

Figure 16. The rebuilt pair passes integrity verification with six active records. The complete CSV comparison also passed.

Physical recovery

A healthy encrypted file is decrypted with the export and Replace procedure in Chapters 4 and 6. Physical recovery is for a damaged file. Restore a verified backup when that is the appropriate way to recover the database.

Choose the recovery route from the file's actual condition.

Condition	Action
Encryption mismatch or FI_CRYPT	Restore the correct configuration before changing files or keys
Missing or damaged index, with a readable datafile	Follow Decide whether to rebuild a missing or damaged index
APPX can export every required record	Preserve and check the complete export; recreate the file with Replace under the intended encryption settings
Damaged encrypted APPX 6.2.1 file	Use a compatible 6.2.1 engine on working copies, then rebuild and verify as described below

Use an APPX 6.2.1 engine with the matching file format, structures, administration configuration, and encryption settings. Do not use a different release's engine as a shortcut for recovering an older physical file.

In the APPX 6.2.1 recovery test, nine active records survived recovery and an index rebuild after a record was deleted. Their complete exported values matched the export made before recovery. The check included a NATIONAL field and its index. Separate copies with a damaged or truncated final record each recovered the other eight records with unchanged values and reported the lost record as an error. A record that cannot be recovered still requires a backup or other trusted source.

1. Stop all access. Preserve the original datafile, index, structure, administration backup, and key records.
2. Work in a separate restricted directory. Choose a new output filename that does not already exist. Never use the source filename as the destination.
3. Run the recovery utility and retain its entire log. The following Unix example uses the example paths described in the previous topic.

```
/opt/appx/appx -recover /srv/appx-recovery/PROSPECT.dat /srv/appx-recovery/PROSPECT-new.dat
```

For Windows PowerShell, substitute your own paths:

```
& 'C:\APPX\appx.exe' -recover 'C:\APPX-Recovery\PROSPECT.dat' 'C:\APPX-Recovery\PROSPECT-new.dat'
```

4. Review the number of active records copied, deleted records removed, and errors found. Investigate skipped or damaged regions; a completed run alone does not establish that all required data was recovered.

5. In a separate test database with matching structures and encryption settings, put the recovered `.dat` under the expected file name. Preserve the old index separately.
6. Select only that test file in Data File Management and run Rebuild key Files. The recovered datafile requires a new matching index. Do not reuse the original `.key` file.
7. Run Verify Integrity of APPXIO Files. Compare counts and application values, and compare a normal APPX export with a known-good export when available.
8. Accept a recovered file only after the data and application checks are satisfactory. Plan any replacement of live data as a separate controlled restoration.

Do not use the recovery utility's

`-c` option as a decryption or CSV-export shortcut. Keep deleted-record outputs, dumps, and recovery logs restricted. They may contain sensitive information.

Extract records flagged as deleted

APPX recovery normally leaves deleted records out of its active-record output. An optional third output pathname requests a separate file containing the deleted records that the utility can extract. It does not automatically add those records back to the application.

The following example is for an encrypted APPX 6.2.1 file with a compatible 6.2.1 engine and the correct encryption configuration. The paths are examples; substitute your own restricted working directory and engine path. Preserve the source and use two new output names.

```
/opt/appx/appx -recover /srv/appx-recovery/PROSPECT.dat /srv/appx-recovery/PROSPECT-
active.dat /srv/appx-recovery/PROSPECT-deleted.dat
```

For Windows PowerShell:

```
& 'C:\APPX\appx.exe' -recover 'C:\APPX-Recovery\PROSPECT.dat' 'C:\APPX-Recovery\PROSPECT-
active.dat' 'C:\APPX-Recovery\PROSPECT-deleted.dat'
```

In the APPX 6.2.1 test with a fixed-length, indexed PROSPECT file, the separate deleted-record output matched every byte of the deleted record's original decrypted contents. All nine surviving active records also retained their complete contents.

For the fixed-length file tested in 6.2.1, the deleted output is unencrypted native consecutive record data. It is not CSV, and some fields retain binary storage. Do not open it as a text export or substitute it for the indexed PROSPECT file. Reading it through APPX requires a separate consecutive-file definition with matching fields, storage types, and record length. Have an APPX designer prepare that definition and a controlled procedure to check the extracted values and deliberately add any wanted record back. Check for duplicate identifiers and related records before reinstating it. The extraction was verified byte for byte; this additional application restoration procedure was not part of the test.

Deleted-record extraction depends on the old contents still being present and undamaged. In the 6.2.1 test, a new APPX record reused the deleted record's storage slot. Recovery then reported zero deleted records and produced an empty deleted output. It could no longer extract the old record from that file. Stop writes and preserve a copy immediately when a deleted record is needed; use an earlier backup if its storage has been reused.

Variable-length deleted-record extraction was not tested. These results do not establish recovery of overwritten data or every form of file damage. Keep the unencrypted deleted output under the same controls as a confidential CSV export.

Test recovery on a disposable copy

Use a disposable APPX 6.2.1 installation with the matching file format, structures, administration configuration, and encryption settings. This exercise checks that recovery preserves the surviving records. Never test against live data.

1. Preserve an encrypted baseline with recognizable records and distinct identifiers. Save its complete CSV export without headings and the matching native file pair.
2. In an independent copy, delete one known record through APPX, then export the remaining records. This is the reference for recovery; it excludes the intentionally deleted record.
3. Run recovery into new active and deleted output filenames as shown in [Extract records flagged as deleted](#). Keep the entire log.
4. Put the active output into a separate test database under its expected file name. Preserve any old index separately. Run Rebuild key Files and Verify Integrity of APPXIO Files for only that file.
5. Export the recovered file with the same heading setting. Compare every field and identifier with the reference, allowing for row order. An export error, missing record, or changed value fails the test even if counts and integrity checks pass.
6. Investigate any export error or changed value before accepting the recovered file. Keep the originals and contact APPX Support when a complete comparison cannot be made.

For a more complete check, repeat the exercise on fresh copies with no deletion and with a record deleted near the beginning, middle, and end of the physical file. Application and index order may differ from physical order. Compare every surviving record in each case.

Compare the deleted output with complete native contents captured before deletion, not just the expected length. For a reuse control, confirm that an APPX insertion actually occupied the old deleted slot before repeating recovery. An insertion alone does not prove reuse.

CHAPTER 8

Routine operation and compatibility

Manage restoration, key changes, APPX ODBC, and Unicode.

Backups, restoration, and another server

Back up the native files with their matching structures and required administration configuration. Protect the backup of System Administration as carefully as the key vault. Keep the exact system passphrase and applicable file and group keys available through the documented recovery route.

Native encrypted-file backups retain their encrypted storage. CSV exports, reports, spreadsheets, application-generated files, and other outputs need their own handling. APPX data file encryption does not automatically encrypt those outputs or protect every other file on the server.

For restoration on another APPX server, follow [Share encrypted files across servers](#). Preserve the actual FMS and path mappings and test the restored copy before allowing application users onto it.

If the administration files are lost or damaged, restore an appropriate administration backup or re-establish the exact settings from the protected key records. Contact APPX Support when the required configuration cannot be restored. Do not assume that physical recovery substitutes for a key backup or that a new passphrase can open an old encrypted file.

Share encrypted files across servers

APPX supports sharing encrypted native datafiles between systems registered to the **same APPX customer**, provided the systems also use the **same system Encryption Passphrase** and the **same effective file encryption key**. All three conditions must be met. Matching customer identity alone is insufficient. These are the conditions published in [Sharing of Encrypted Files](#).

The servers can have separate APPX registrations. The requirement is the same APPX customer identity on those registrations; their full registration records do not have to be identical.

For example, a production server and a disaster-recovery server belonging to the same APPX customer can be configured to read the same encrypted backup. The native files can remain encrypted during transfer and restoration.

1. Confirm that both APPX systems are registered to the same APPX customer. Ask APPX Support to confirm the customer identity if the registrations are uncertain.
2. Confirm that the destination engine supports the physical file format and has the matching application structures.
3. On a separate destination test installation, set the exact system passphrase and effective file or group keys from the restricted vault. Preserve the intended FMS and destination path mappings.
4. Make a consistent copy while source-file access is stopped. Transfer the complete `.dat` and `.key` pair through an approved protected channel, and preserve the source copy.

5. Open the destination copy through APPX, verify integrity, compare record counts and complete sample values, and check application totals before accepting it.

Do not change an existing destination system's passphrase merely to open an incoming file. Such a change affects that system's other encrypted files; follow [Change a key or the system passphrase](#) when a system-wide conversion is required.

When two systems should remain separate, use different system passphrases and file keys, as recommended in the sharing reference. Keep the recovery records for each system clearly identified in the vault.

Change a key or the system passphrase

A file or group key change is another conversion. Export under the old working settings, preserve and check the exports, change the affected settings, and import with Replace. A group-key change requires every affected member to be included, across applications and databases.

A system passphrase change has wider scope:

1. Stop users, jobs, API sessions, and ODBC activity for the entire affected APPX system.
2. Back up the data and administration configuration and preserve the old passphrase and keys.
3. Use the List of FMS Groups and Files using Encryption report on Set/Change Pass Phrase to help inventory the configuration. Check the inventory against actual file and group usage.
4. Export every encrypted datafile in every affected database under the old working configuration. Check all exports before changing the passphrase.
5. Set the new system passphrase and reset the affected file and group keys under it. Retain the exact new values securely.
6. Import every affected file with Replace, then verify integrity, counts, records, and application totals.
7. Make and test a new backup. Keep the old key records for older backups.

Do not perform a partial system passphrase change while leaving other encrypted databases under the old configuration.

APPX ODBC and external database interfaces

APPX 6.2.1 can read encrypted native APPXIO files through APPX ODBC. The tested 6.2.1 configuration returned the expected values from an encrypted file, including a unique test value after a clean restart. Before converting files used by Excel, reports, or other ODBC clients, complete [Check ODBC access before converting a file](#). See [ODBC access in the 6.2.1 test](#) for the tested configuration and results.

The APPX side of the connection needs the correct administration context, registration, and effective encryption settings. The standard desktop ODBC DSN has no additional APPX file-encryption key field. Retain the applicable APPX and ODBC access controls. Protect client query results and exports whenever a query succeeds. File encryption is separate from securing the client connection and server network access.

Do not confuse these two directions:

Connection	Where the data resides	APPXIO encryption
Excel or another client reads native APPXIO through APPX ODBC	Native APPXIO files on the APPX side	Native storage can be encrypted; encrypted-file ODBC access requires its own compatibility check
APPX accesses SQL Server through an external database interface, including ODBC FMS	SQL Server tables	Does not encrypt those tables
APPX accesses Oracle through its Oracle interface	Oracle tables	Does not encrypt those tables

Verify the exact gateway, driver, client, permissions, and character handling used at your site. Use the applicable [APPX ODBC cookbook](#) for that connection's setup.

Check ODBC access before converting a file

Use a separate APPX 6.2.1 test copy. COUNTRY is convenient because the APPX ODBC cookbook imports its definition as DMO_DMO_COUNTRY. The 6.2.1 test had eight records and nine SQL columns. Your data and counts may differ.

1. With COUNTRY unencrypted and ODBC working, run the three queries below. Save all returned values, including every column of the full-file query. Close the client connections.
2. Back up the file pair, matching structures, and administration configuration. Export COUNTRY through APPX and preserve the complete CSV.
3. Follow [Set the file key and replace the file](#), substituting COUNTRY for PROSPECT and using a new file key. Select only COUNTRY. Use no inherited group key for this independent test.
4. Confirm the physical ENCRYPTED flag on a complete working copy as described in [Analyze a working copy](#). Export the encrypted file through APPX and compare the complete CSV with the original.
5. Open a fresh ODBC connection using the same DSN and login. Repeat the three queries. Record errors or a failure to finish, as well as successful results. Compare complete values rather than relying only on COUNT.
6. Close the test connections. Complete the [file decryption procedure](#), again substituting COUNTRY. Confirm that the physical encryption flag is absent, then open a fresh ODBC connection and repeat the queries for the final control.

```
SELECT COUNT(*) AS country_count FROM DMO_DMO_COUNTRY;
SELECT * FROM DMO_DMO_COUNTRY ORDER BY name;
SELECT name, no_words FROM DMO_DMO_COUNTRY WHERE name = 'Canada';
```

Compare complete query results with both the saved ODBC baseline and the native APPX export. Close and reopen connections between states so the check reads the current file. Resolve connection or character-handling problems before relying on client output.

ODBC access in the 6.2.1 test

APPX 6.2.1 successfully read the encrypted native COUNTRY file through ODBC in this test. The configuration used the 64-bit Windows APPX 6.2.1.0 engine, matching 6.2.1.0 gateway, supplied 64-bit Dharma 14.01 driver, and 64-bit Windows PowerShell 5.1. COUNTRY was a native format-3 file with eight records and nine SQL columns. The test ran in an isolated VM with its network adapter disconnected.

The queries were:

```
SELECT COUNT(*) AS country_count FROM DMO_DMO_COUNTRY;
SELECT * FROM DMO_DMO_COUNTRY ORDER BY no_words, name;
SELECT no_words FROM DMO_DMO_COUNTRY ORDER BY no_words;
```

Check	Result
Unencrypted controls with a fresh gateway for each case	All three queries passed
Native APPX export while encrypted	A newly created CSV matched the original byte for byte; the physical ENCRYPTED flag was present
ODBC while encrypted	Count returned 8; the full-row query returned all eight rows and nine columns; all 72 returned values matched the ODBC baseline
After decryption	All three queries passed; returned values and the new native CSV matched their baselines
Unique value after re-encryption and restart	ODBC returned the expected numeric value 7 and audit value JPN from the encrypted file

For the final proof, Canada's no_words value was changed from 2 to 7 through APPX CSV Replace in the disposable test copy. A native export confirmed the change while the file remained encrypted. After a clean VM restart and another physical encryption check, this fresh ODBC query returned exactly one row containing 7 and JPN:

```
SELECT no_words, added_by FROM DMO_DMO_COUNTRY WHERE no_words = 7;
```

The original test data contained no no_words value of 7. This extra check established that ODBC read the marked encrypted file. Use a suitable unique value for your own test dictionary and data.

The test also encountered intermittent connection-setup stalls before SQL execution, including with encryption disabled. Fresh unencrypted controls and the marker query after restart succeeded. If connection setup stalls at your site, resolve that separately and repeat the check with fresh connections.

Check complete text values against native APPX. The tested ODBC configuration truncated some text even with encryption disabled; encryption and decryption did not change that behavior. Matching ODBC outputs across encryption states alone do not establish correct character handling. The numeric and audit marker values above matched their native values.

Unicode in APPX 6.2.1

APPX 6.2.1 encrypts the stored record bytes; it does not apply a different encryption procedure to non-ASCII text. No separate encryption setting is required for Unicode fields.

Keep the APPX engine, application dictionary, and physical file structure appropriate to the data. An encryption conversion is not a conversion between an older character representation and a Unicode representation. Handle an application or file-format upgrade through its normal upgrade and restructure procedures.

APPX 6.2.1 CSV export converts character data to UTF-8. When opening that CSV in another program, select UTF-8 and check complete field values. Include accents and other scripts actually used in the application. Keep the original unedited export for restoration or comparison.

The 6.2.1 encrypted-file recovery test included non-ASCII values in a NATIONAL field and its index. After recovery and an index rebuild, the UTF-8 CSV matched the pre-recovery export byte for byte, including those characters.

Verify character handling when upgrading or changing clients. ODBC Unicode behavior also depends on the exact driver and client; file encryption does not replace those checks.

When a check fails

Symptom	Check before changing data
Encryption mismatch or FI_CRYPT	Exact system passphrase, effective group and file keys, registration, and administration tree
A file remains encrypted after a planned decryption	Replacement import completed; no individual key or inherited group key remains effective
A file remains unencrypted after a planned encryption	Eligible organization, saved passphrase and key settings, correct database and application selection, Replace rather than Merge
Fewer records after import	Correct CSV and heading choice, complete export, import errors, file selection, duplicate-key errors, and matching structures
Missing index or failed index verification	Follow Decide whether to rebuild a missing or damaged index ; rebuild from a copy of the original datafile before considering physical recovery
Different path from the example	APPXPATH, APPX_OSA_PATH, file and group path overrides, and the actual export location
CSV is readable although the APPXIO file is encrypted	Expected export behavior; protect the CSV separately

Keep the failing log and original copies. Correct an identified configuration problem before repeating a destructive import or changing keys.

APPENDIX A

Reference and completion checks

Keep a completion record and find the supporting references.

Desktop Client keys

In the Desktop Client used here, F3 is SELECT, F5 is Next Record, and F8 is END. Keyboard mappings can differ. The File Selection screen identifies SELECT as the command that toggles one file. The **Select** button opens a selection-criteria dialog; it is a different action.

Conversion completion record

Keep the following information with the maintenance record, and keep the secret values themselves in the restricted vault.

- Database and every related application included in the conversion.
- File inventory, organizations, FMS types, paths, and group assignments.
- Export locations, heading choices, record counts, and protected backup locations.
- Vault references and key versions, including the previous versions needed by retained backups.
- Import disposition and results for every file.
- Physical encryption-state confirmation and integrity results.
- Record comparisons, application checks, and any CSV comparisons.
- The person responsible for restoration and the tested route to the protected recovery material.

For the sample data illustrated in this manual, the nine-file conversion inventory totals 125 records. Record and compare your own counts for the separate PROSPECT exercise and for every file in the DMO database exercise.

Further reading

- [APPXIO Data Encryption](#): the original feature reference, including system passphrases, file and group settings, and cross-server requirements.
- [APPX Library](#): System Management and Application Design manuals and the versioned APPX ODBC cookbooks.
- [1Password vault access](#): sharing a vault with the appropriate people.
- [1Password organization recovery plan](#): preparing recovery access before an emergency.