Suite BKPIFS

Native Direct Backup to IFS for IBM i

Minimum Release V7R2M0

INDEX

Suite BKPIFS.....1

What is the BKPIFS Suite?.....4
Key Infrastructure Benefits.....5
How to Integrate It into Your Business Workflow.....6
Prerequisites for Activation.....6
Quick Access via Operational Menu (GO BKPIFS).....7
License and Activation Key.....8

SAVLIBIFS (Save Library to IFS).....10
Restrictions and Prerequisites.....10
Command Parameters.....10
Practical Usage Examples.....11
Error Handling and Escape Messages.....11

SAVOBJIFS (Save Object to IFS).....13
Restrictions and Prerequisites.....13
Command Parameters.....13
Practical Usage Examples.....14
Error Handling and Escape Messages.....15

SAVCHGIFS (Save Changed Objects to IFS).....16
Restrictions and Prerequisites.....16
Command Parameters.....16
Practical Usage Examples.....17
Error Handling and Escape Messages.....18

SAVLNKIFS (Save Links to IFS).....19
Restrictions and Prerequisites.....19
Command Parameters.....19
Practical Usage Examples.....20
Error Handling and Escape Messages.....20

RSTLIBIFS (Restore Library from IFS)..22
Restrictions and Prerequisites.....22
Command Parameters.....22
Practical Usage Examples.....23
Error Handling and Escape Messages.....23

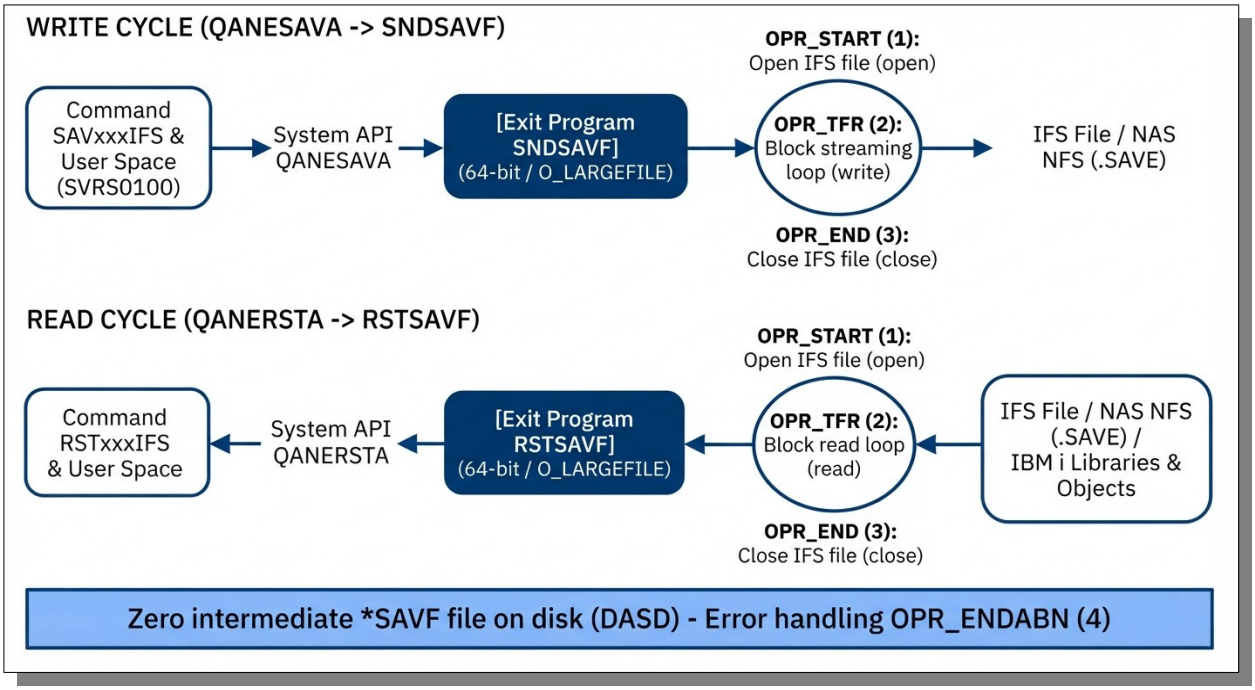
RSTOBJIFS (Restore Object from IFS). 24
Restrictions and Prerequisites.....24
Command Parameters.....24
System Behavior and Security.....25
Practical Usage Examples.....25
Error Handling and Escape Messages.....26

RSTLNKIFS (Restore Links from IFS)...27
Restrictions and Prerequisites.....27
Command Parameters.....27
Practical Usage Examples.....27
Error Handling and Escape Messages.....

Introduction and Operational Integration Guide

What is the BKPIFS Suite?

The **BKPIFS** suite is an integrated set of native commands for the IBM i (iSeries) environment designed to **perform save and restore operations directly on the Integrated File System (IFS)**.
Unlike traditional methods that require creating intermediate Save Files (*SAVF) on disk or using physical/virtual storage media, the BKPIFS suite acts as a true "virtual streaming device." By leveraging official IBM system APIs **QANESAVA** (for saving) and **QANERSTA** (for restoring), the product reads and writes data blocks in real time to and from the IFS.



Key Infrastructure Benefits

- **Zero impact on mass storage (DASD):** Because no intermediate `*SAVF` files are created in `QGPL` or temporary libraries, local disk space consumed during backup is zero. Data streams directly to the target IFS directory.
- **Total restore flexibility:** The suite guarantees absolute granularity. If an entire library is saved using the `SAVLIBIFS` command, the administrator can restore even a single object by running an `RSTOBJIFS` command pointed directly at that stream file.
- **Clean, self-contained architecture:** All components (commands, control programs, and Help panels) are completely contained within the `BKPIFS` library. No invasive installations or operating system modifications are required.

How to Integrate It into Your Business Workflow

The `BKPIFS` suite is designed to fit seamlessly into any data protection strategy, whether manual or automated:

1. **Integration with Job Schedulers (e.g., Advanced Job Scheduler or Robot/SAVE):** Commands such as `SAVLIBIFS` or `SAVLNKIFS` can replace or complement standard `SAVLIB` commands in end-of-day CL programs.
2. **Cloud and Network Storage Strategies (NAS/SAN):** By saving directly to the IFS, you can target directories mounted via **NFS (Network File System)** or **QNTC (Windows/Samba shares)**. This writes the iSeries backup directly to external storage or a Cloud-synced folder, eliminating the need for FTP scripts or third-party tools to export `*SAVF` files.
3. **Fast Test and Development Environments:** Allows developers to duplicate or freeze the state of a library before a software update with a single command, placing the file in an IFS staging folder and restoring it on the fly in case of a rollback.

Prerequisites for Activation

To begin using the suite commands, the only preliminary step required is adding the library to your Library List:

```
ADDLIBLE LIB(BKPIFS) POSITION(*LAST)
```

Once this command is executed, all tools (`SAVLIBIFS`, `RSTLIBIFS`, etc.) are immediately available for interactive sessions or batch jobs.

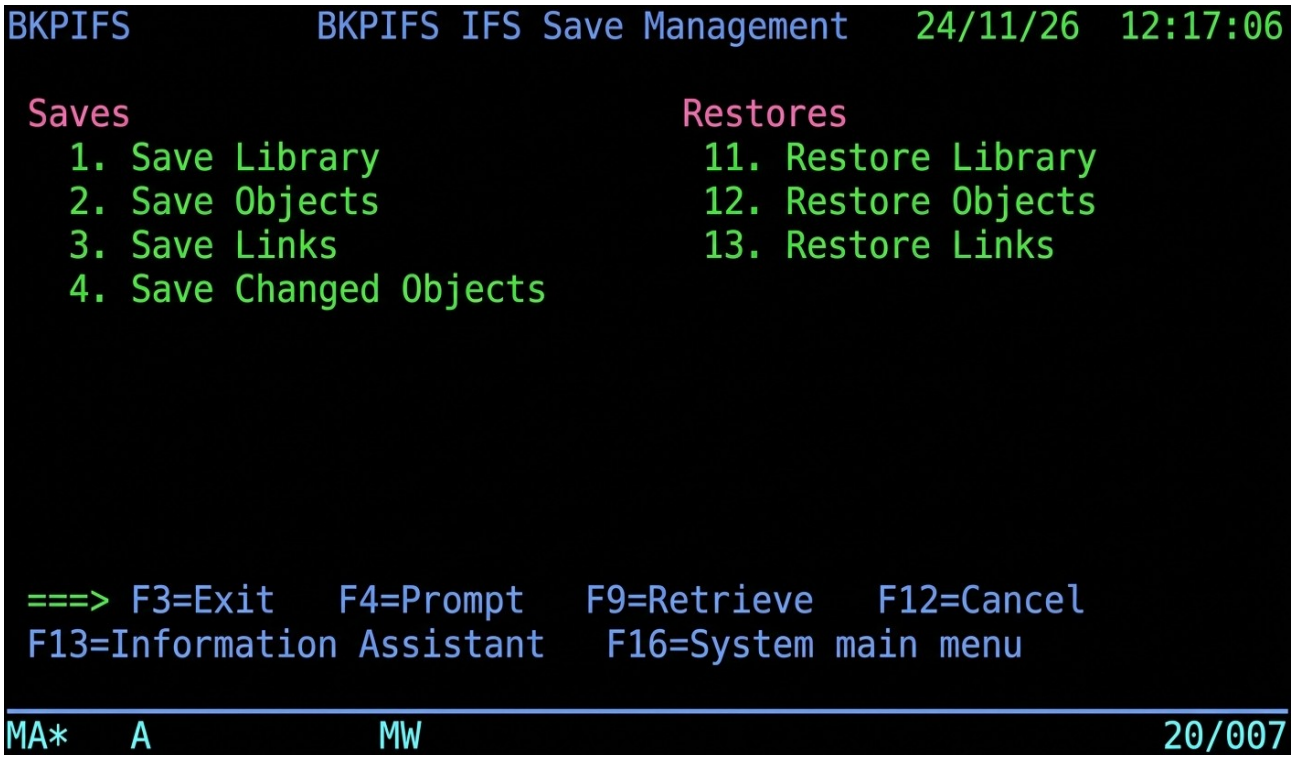
Quick Access via Operational Menu (`GO BKPIFS`)

In addition to direct execution from the command line or within CL programs, the suite provides a dedicated interactive menu to simplify manual operations for system operators. After adding the library to the library list, simply enter:

```
GO BKPIFS
```

The main menu (**BKPIFS Gestione Salvataggi su IFS / IFS Save Management**) organizes all suite functions into two operational sections with automatic parameter prompting (F4):

- **Saves (Options 1–4):** Quick access to `SAVLIBIFS` (1), `SAVOBJIFS` (2), `SAVLNKIFS` (3), and `SAVCHGIFS` (4).
- **Restores (Options 11–13):** Quick access to `RSTLIBIFS` (11), `RSTOBJIFS` (12), and `RSTLNKIFS` (13).



License and Activation Key

The **BKPIFS** suite is distributed under a flexible licensing model designed to ensure continuous operation for primary backups:

- **Free Command (SAVLIBIFS):** Full library save functionality is freely available without any activation key.
- **Pro Commands (Activation Key Required):** All other commands in the suite (RSTLIBIFS, SAVOBJIFS, RSTOBJIFS, SAVCHGIFS, SAVLNKIFS, RSTLNKIFS) require a valid authorization key to run.

How to Request and Enter the Key

To obtain an activation key, contact **info@abacosoft.it** or your software provider, supplying your IBM i machine serial number (retrieved via DSPSYSVAL QSRLNBR).
Once received, the key must be stored inside the **BKPIFSAUT** data area, located in the product library (BKPIFS), using the standard system command:

```
CHGDTAARA DTAARA(BKPIFS/BKPIFSAUT) VALUE('CHIAVE_DI_ATTIVAZIONE')
```

(Note: The BKPIFSAUT data area must always reside in the BKPIFS library. For annual license renewals, simply update the value using the same command).

License Terms and Intellectual Property

Purchasing an annual subscription (or a perpetual license) grants the customer solely a **non-exclusive, non-transferable right-to-use license tied to a single machine serial number (QSRLNBR)** for the agreed validity period.
The software is provided strictly in **compiled object format**: under no circumstances does the license transfer intellectual property ownership or grant any right to receive the source code, which remains the sole property of **Abacosoft di Mario Pippia**. Redistribution, sublicensing to third parties, or reverse engineering of the suite components is strictly prohibited.

System Messages and Expiration Management (*MSGF)

When executing Pro Edition commands, the BKPIFS suite validates the key stored in BKPIFS/BKPIFSAUT and logs the following custom messages in the Job Log:

Message ID	Severity	Message Text	Meaning and Behavior
BKP0001	0	*BKPIFS scade il &1.*	**Expiration warning.** Displays the current key expiration date starting 7 days prior to expiration. The command runs normally
BKP0002	30	*Attenzione: BKPIFS è scaduto il &1*	**Expired license block.** The key validity date has passed and command execution is halted. Please renew your license.
BKP0003	50	*ATTENZIONE! LICENZA BKPIFS NON VALIDA!*	**Blocco operativo.** La chiave è assente o errata. L'esecuzione del comando viene interrotta.**Operational block.** The key is missing or invalid for this machine serial number. Command execution is halted.

(Administrator Note: In automated CL programs, you can trap license blocks by monitoring BKP0002 and BKP0003 via MONMSG MSGID(BKP000x).

Legal Notes and Acknowledgments (Open Source / Copyleft)

The **BKPIFS** suite—including its commands, menu interface, online help panels (UIM), license key protection system, and architectural extensions for 64-bit large file support (O_LARGEFILE)—is developed and distributed by **Abacosoft di Mario Pippia** (© All rights reserved).
Certain foundational streaming routines utilizing the IBM system APIs QANESAVA and QANERSTA are derived from public domain/copyleft code originally published by **Carsten Flensburg**, appropriately adapted and extended to ensure compatibility with large files and modern production environments.

SAVLIBIFS (Save Library to IFS)

The **SAVLIBIFS** command saves a copy of one or more libraries **directly into a stream file (STMF) residing in the Integrated File System (IFS)**. Leveraging the IBM system API QANESAVA, the command does not require intermediate Save File (*SAVF) disk space and automatically updates the last save date, time, and location in the library description.

Restrictions and Prerequisites

- **System Authorities:** Requires ***SAVSYS** special authority.
- **Alternative Authorities:** Without *SAVSYS, the user must possess:
- Object existence authority (***OBJEXIST**) for every object to be saved.
- Execute authority (***EXECUTE**) for the containing library.
- *Note:* If not run under *SAVSYS, only objects for which the user holds proper authority will be saved.

Command Parameters

Parameter (Keyword)	Description and Accepted Values	Type / Default
LIB	**Library.** Specifies the name of the system library to back up.	*NAME *Required*
STMF	**IFS File.** Specifies the absolute path and name of the target stream file in the IFS (e.g., /backup/lib/company.bak).	*PATH *Required*
REPLACE	**Replace Existing File.** Controls behavior if the target IFS file already exists. • *NO: If the file exists, the command halts with an error. • *YES: Overwrites the existing IFS file with the new backup.	Predefined options Default: ***NO**
TGTRLS	**Target Release.** Specifies the minimum operating system release on which the data can be restored. • *CURRENT: Current release installed on the machine. • *PRV: Allows restoring on the previous IBM i release.	Predefined options Default: ***CURRENT**
UPDHST	**Update History.** Specifies whether to record backup information in the library description. • *YES: Stores current date/time on the library descriptor. • *NO: Does not alter library history data.	Predefined options Default: ***YES**
SAVACT	**Save While Active.** Enables backing up objects while in use by other jobs. • *NO: Objects locked by applications are not saved. • *LIB / *SYNCLIB / *SYSDFN: Manages synchronization checkpoints at the library or system level.	Predefined options Default: ***NO**
SAVACTWAIT	**Save Active Wait Time.** Structured in 3 elements to control lock wait times in seconds: 1. Object locks (Default: 120). 2. Pending record changes (Default: *LOCKWAIT). 3. Other pending changes (Default: *LOCKWAIT).	Multiple elements Default: **120** ***LOCKWAIT** ***LOCKWAIT**

SAVACTMSGQ	**Save Active Message Queue.** Specifies the message queue to receive checkpoint synchronization notifications. Qualifier 1: Queue name or `*NONE`. Qualifier 2: Library (Default `*LIBL`).	Qualified queue Default: **`*NONE`**
SAVFDTA	**Save Save File Data.** Specifies whether to save the actual contents of Save Files (`*SAVF`) within the library or only their empty structure. Values: `*YES`, `*NO`.	Predefined options Default: **`*YES`**
SPLFDTA	**Save Spool File Data.** Determines whether to include spooled file data from output queues associated with the library. Values: `*NONE`, `*ALL`.	Predefined options Default: **`*NONE`**
QDTA	**Queue Data.** Defines whether to save the data contents inside Data Queues (`*DTAQ`). Values: `*NONE`, `*DTAQ`.	Predefined options Default: **`*NONE`**
OUTPUT	**Output.** Chooses whether to generate a printed report of saved objects. Values: `*NONE`, `*PRINT` (generates `QPSAVOBJ` spool file).	Predefined options Default: **`*NONE`**

Practical Usage Examples

Example 1: Standard save with safety overwrite protection

Performs a full backup of the `DATI_PROD` library creating `dati_prod.bak` in the `/IFSBACKUP/` IFS directory. If the file already exists, the operation stops safely.

```
SAVLIBIFS LIB(DATI_PROD) STMF('/IFSBACKUP/dati_prod.bak') REPLACE(*NO)
```

Example 2: Hot backup (Save While Active) and forced overwrite

Saves the `LIBRERIA_A` library forcing overwrite of the existing IFS file, enabling `*SYNCLIB` mode to avoid blocking users during daily production and setting a 60-second maximum lock wait.

```
SAVLIBIFS LIB(LIBRERIA_A)
     STMF('/IFSBACKUP/lib_a.bak')
     REPLACE(*YES)
     SAVACT(*SYNCLIB)
     SAVACTWAIT(60 *LOCKWAIT *LOCKWAIT)
```

Error Handling and Escape Messages

The command returns controlled errors via `*ESCAPE` messages originating from file system monitoring and the OS save subsystem:

- **`CPF9801`**: Library or resource not found on the system.

- `CPF9802`: Insufficient authority for the library or target IFS directory.
- `CPF9820`: Security violation accessing the IFS directory tree.
- `CPF3700` **series (indirect)**: Hardware issues or exclusive locks encountered during object serialization via `QANESAVA` APIs.

SAVOBJIFS (Save Object to IFS)

The **SAVOBJIFS** command saves a single object, a group of specific objects, or entire generic sets belonging to a specific library, **writing them directly into a stream file (STMF) in the IFS**. The stream file is dynamically created during processing via the QANESAVA API. Upon completion, each object's descriptor is updated with the last save date and time (viewable via DSPOBJD), unless history updating is disabled.

Restrictions and Prerequisites

- **Pro Edition:** Command reserved for the full version of the BKPIFS suite (requires a valid activation key in the BKPIFSAUT data area).
- **System Authorities:** Requires ***SAVSYS** special authority.
- **Alternative Authorities:** Without *SAVSYS, the user must possess:
- Object existence authority (***OBJEXIST**) for every object included in the backup.
- Execute authority (***EXECUTE**) for the library containing the objects.
- *Note:* Objects for which the user lacks required authorities will be skipped during the backup.

Command Parameters

Parameter (Keyword)	Description and Accepted Values	Type / Default
OBJECT	**Objects.** Specifies the names of the objects to save. • *ALL: Includes all objects matching the requested type. • *object-name*: Specific name of a single object. • *generic-name**: Includes groups of objects using a trailing asterisk (e.g., FATT*). Supports up to 300 repetitions.	Single value or list *Required*
LIB	**Library.** Indicates the system library where the objects to save reside. Generic library names are not accepted.	*NAME *Required*
OBJTYPE	**Object Types.** Specifies the object types to include (e.g., *FILE, *PGM, *CMD). • *ALL: Saves all object types matching the OBJECT filter. Supports up to 300 different types in the same command.	Single value or list Default: ***ALL**
STMF	**Stream File.** The absolute path of the IFS file to create and use for storing the save stream (e.g., /backup/oggetti/prog.bak). Inherits permissions from the parent directory.	*PATH *Required*
TGTRLS	**Target Release.** Defines the minimum IBM i version on which you intend to restore the objects. • *CURRENT: Same as current system release or later. • *PRV: Allows restoring on the immediately preceding release.	Predefined options Default: ***CURRENT**
UPDHST	**Update History.** Specifies whether to update the object's save history. Important for establishing the correct starting point for journal apply commands (APYJRNCHG). Values: *YES, *NO.	Predefined options Default: ***YES**
SAVACT	**Save While Active.** Enables backing up objects even if locked	Predefined options

	or in use by other jobs. • *NO: Busy objects are not saved. • *LIB / *SYNCLIB / *SYSDFN: Synchronizes checkpoints at the library or system level to ensure relational consistency.	Default: ***NO**
SAVACTWAIT	**Save Active Wait Time.** Structured in 3 elements for managing record/object lock queues: 1. Object locks: Seconds to wait before skipping the object (Default 120 or *NOMAX). 2. Record changes: Wait time for commit boundaries (Default *LOCKWAIT, *NOCMTBDY for partial transactions). 3. Other changes: Wait time for DDL changes (Default *LOCKWAIT).	Multiple elements Default: **120 *LOCKWAIT *LOCKWAIT**
SAVACTMSGQ	**Save Active Message Queue.** Queue to receive checkpoint completion notifications (useful to promptly release external applications). Qualifier 1: Queue name or *NONE. Qualifier 2: Library (*LIBL or *CURLIB).	Qualified queue Default: ***NONE**
OUTPUT	**Output.** Generates a list of processed objects in the current job's spool. Values: *NONE, *PRINT (produces QPSAVOBJ spool file).	Predefined options Default: ***NONE**

Practical Usage Examples

Example 1: Targeted program backup

Saves all programs (*PGM objects) starting with the letters "FT" contained in the CONTAB library, writing the stream to programmi_ft.bak.

```
SAVOBJIFS OBJECT(FT*) LIB(CONTAB) OBJTYPE(*PGM)
     STMF('/backup/oggetti/programmi_ft.bak')
```

Example 2: Multi-object hot backup (Save Active)

Saves specific physical and logical files from the RISORSE library, waiting up to 60 seconds for objects to become available and allowing checkpoint completion even with active transactions.

```
SAVOBJIFS OBJECT(ANAGR00F ORDINI01L)
     LIB(RISORSE)
     OBJTYPE(*FILE)
     STMF('/backup/oggetti/tabelle.bak')
     SAVACT(*SYNCLIB)
     SAVACTWAIT(60 *NOCMTBDY *LOCKWAIT)
```

Error Handling and Escape Messages

The command returns controlled errors via `*ESCAPE` messages originating from file system monitoring and the OS save subsystem:

- **`CPF9801`**: Library or target IFS directory not found on the system.
- **`CPF9802`**: Insufficient authority for the source library, objects to save, or IFS folder.
- **`CPF9820`**: Security violation accessing the IFS directory tree.
- **`CPF3771`** / **`CPF3773`**: Some or all specified objects were not saved (e.g., object not found or locked in exclusive use).
- **`CPF3700`** **series**: Errors propagated during the `QANESAVA` save session (e.g., stream file already exists in the IFS).

SAVCHGIFS (Save Changed Objects to IFS)

The **SAVCHGIFS** command performs an incremental backup of all objects (or a specific selection) **that have been created or modified since a specified reference date and time**, writing the data directly into a stream file (STMF) in the IFS.
This command is especially suited for optimizing nightly or intraday backup windows: it allows you to quickly save only the changes made after the last full backup (SAVLIBIFS), reducing processing time and storage space consumed on the target destination.

Restrictions and Prerequisites

- **Pro Edition:** Command reserved for the full version of the BKPIFS suite (requires a valid activation key in the BKPIFSAUT data area).
- **System Authorities:** Requires ***SAVSYS** special authority, or object existence authority (***OBJEXIST**) for each modified object and execute authority (***EXECUTE**) on the source library.

Command Parameters

Parameter (Keyword)	Description and Accepted Values	Type / Default
OBJECT	**Objects.** Specifies the names of the objects to check and save if modified. • *ALL: Checks all objects in the library. • *object-name* / *generic-name**: Specific names or groups with a trailing asterisk (up to 300 repetitions).	Single value or list *Required*
LIB	**Library.** Indicates the system library in which to search for changed objects.	*NAME *Required*
OBJTYPE	**Object Types.** Filters the object types to save (e.g., *FILE, *PGM, *DTAARA). • *ALL: Includes all object types matching the OBJECT filter (up to 300 repetitions).	Single value or list Default: ***ALL**
STMF	**IFS File.** The absolute path and name of the target stream file in the IFS (e.g., /backups/inc/dati_chg.bak).	*PATH *Required*
REPLACE	**Replace.** Controls overwriting of the target stream file. • *NO: Halts the command if the file already exists. • *YES: Overwrites the pre-existing file.	Predefined options Default: ***NO**
REFDATE	**Reference Date.** Specifies the date from which to check for object modifications. • *SAVLIB: Automatically uses the date of the last full library save. • *date*: Specific date in system format (e.g., 11/15/2026).	Date or special value Default: ***SAVLIB**
REFTIME	**Reference Time.** Specifies the exact time paired with REFDATE. • *NONE: Covers the entire reference day (or the exact time of the last SAVLIB if REFDATE(*SAVLIB) is used). • *time*: Specific time (e.g., 180000).	Time or special value Default: ***NONE**

TGTRLS	**Target Release.** Minimum operating system release on which data can be restored (`*CURRENT` or `*PRV`).	Predefined options Default: **`*CURRENT`**
UPDHST	**Update History.** Indicates whether to update the last save date and time in the saved objects' attributes (`*YES`, `*NO`).	Predefined options Default: **`*YES`**
SAVACT	**Save While Active.** Enables backing up objects in use by other jobs (`*NO`, `*LIB`, `*SYNCLIB`, `*SYSDFN`).	Predefined options Default: **`*NO`**
SAVACTWAIT	**Save Active Wait Time.** Wait time in seconds for object locks, pending record changes, and other DDL changes.	Multiple elements Default: **`120`** **`*LOCKWAIT`** **`*LOCKWAIT`**
SAVACTMSGQ	**Save Active Message Queue.** Qualified message queue to receive checkpoint completion notifications.	Qualified queue Default: **`*NONE`**
OUTPUT	**Output.** Specifies whether to generate a detailed listing of saved objects (`*NONE`, `*PRINT`, `*OUTFILE`).	Predefined options Default: **`*NONE`**

Practical Usage Examples

Example 1: Automatic incremental save relative to the last full backup

Saves all objects in the `PROD_DATI` library that were created or modified after the last full library backup (`*SAVLIB`), overwriting the incremental file on the NAS.

```
SAVCHGIFS OBJECT(*ALL)
     LIB(PROD_DATI)
     STMF('/QNAP_BACKUP/inc/prod_dati_chg.bak')
     REPLACE(*YES)
     REFDATE(*SAVLIB)
```

Example 2: Intraday save of database files modified after 1:00 PM

Saves only `*FILE` objects modified starting from 13:00 on 16/11/2026, enabling *Save Active* mode to avoid disrupting active users.

```
SAVCHGIFS OBJECT(*ALL)
     LIB(PROD_DATI)
     OBJTYPE(*FILE)
     STMF('/QNAP_BACKUP/inc/pomeriggio.bak')
     REPLACE(*YES)
     REFDATE('16/11/26')
     REFTIME(130000)
     SAVACT(*SYNCLIB)
```

Error Handling and Escape Messages

- **`CPF9801`**: Library or target IFS directory not found.
- **`CPF9802`**: Insufficient authority on the source library or IFS path.
- **`CPD3726`**: `SAVACTWAIT` or `SAVACTMSGQ` parameters not valid when `SAVACT(*NO)` is specified.
- **`CPF3700` series**: Errors propagated during the `QANESAVA` save session (e.g., no changed objects found or active exclusive lock).

SAVLNKIFS (Save Links to IFS)

The **SAVLNKIFS** command saves stream files, directories, and entire folder trees residing in the Integrated File System (IFS) **directly into a single stream file (STMF)**. Leveraging the QANESAVA system APIs, it allows archiving IFS resources (including network shares or application directories) without creating intermediate Save Files (*SAVF) on disk.

Restrictions and Prerequisites

- **Pro Edition:** Command reserved for the full version of the BKPIFS suite (requires a valid activation key in the BKPIFSAUT data area).
- **System Authorities:** Requires ***SAVSYS** special authority, or specific read authority (***R**) on each object to be saved and execute/traverse authority (***X**) on all directories in the path.

Command Parameters

Parameter (Keyword)	Description and Accepted Values	Type / Default
OBJ	**Object.** Specifies the absolute path of the IFS object or directory to save (up to 30 different paths). • *path*: Full IFS path (e.g., '/Company/Documents').	IFS Path (*PNAME) *Required*
STMF	**IFS File.** The absolute path and name of the target stream file where the backup stream will be written (e.g., '/QNAP_BACKUP/ifs_doc.bak').	IFS Path (*PNAME) *Required*
REPLACE	**Replace.** Controls overwriting if the target stream file already exists. • *NO: Halts the command with an error. • *YES: Overwrites the pre-existing stream file.	Predefined options Default: ***NO**
SUBTREE	**Directory Subtree.** Specifies how subdirectories are handled. • *ALL: Saves the entire directory tree (directories, subfolders, and all files). • *DIR: Saves only objects in the main directory without descending into subfolders. • *NONE: Saves only the directory object itself. • *OBJ / *STG: Advanced options for saving only objects or storage.	Predefined options Default: ***ALL**
TGTRLS	**Target Release.** Indicates the minimum IBM i operating system level on which data can be restored (*CURRENT or *PRV).	Predefined options Default: ***CURRENT**
UPDHST	**Update History.** Specifies whether to update the last save date and time in the IFS object attributes (*NO, *YES, *SYS, *PC).	Predefined options Default: ***NO**
SAVACT	**Save While Active.** Allows saving stream files currently in use by other jobs (*NO, *LIB, *SYNCLIB, *SYSDFN).	Predefined options Default: ***NO**
SAVACTWAIT	**Save Active Wait Time.** Wait time in seconds for releasing object locks, pending record changes, and other DDL	Multiple elements Default: **120**

	changes.	***LOCKWAIT** ***LOCKWAIT**
SAVACTMSGQ	**Save Active Message Queue.** Qualified message queue to receive checkpoint completion notifications.	Qualified queue Default: ***NONE**
SYSTEM	**System.** Specifies which file systems to include based on location (`*LCL` local only, `*RMT` remote only, `*ALL` both).	Predefined options Default: ***LCL**
PVTAUT	**Private Authorities.** Determines whether to save private authorities along with the objects (`*YES`, `*NO`).	Predefined options Default: ***YES**
OUTPUT	**Output.** Specifies whether to generate a job spool report listing processed objects (`*NONE`, `*PRINT`).	Predefined options Default: ***NONE**

Practical Usage Examples

Example 1: Full backup of an application folder in the IFS

Saves the entire contents of the `/WebApps/Portale` directory and all its subdirectories into the `portale.bak` stream file on the mounted NAS, including private authorities.

```
SAVLNKIFS OBJ('/WebApps/Portale')
     STMF('/QNAP_BACKUP/ifs/portale.bak')
     REPLACE(*YES)
     SUBTREE(*ALL)
     PVTAUT(*YES)
```

Example 2: Hot backup of multiple directories (Save Active)

Simultaneously saves two separate directories, waiting up to 60 seconds for any files locked by users and generating a printed report.

```
SAVLNKIFS OBJ('/FattureElettroniche' '/AllegatiPDF')
     STMF('/QNAP_BACKUP/ifs/documenti_fe.bak')
     REPLACE(*YES)
     SAVACT(*LIB)
     SAVACTWAIT(60 *LOCKWAIT *LOCKWAIT)
     OUTPUT(*PRINT)
```

Error Handling and Escape Messages

- **CPFA0A9**: Source IFS object or path not found.
- **CPFA09C**: Insufficient authority on the source IFS object or target directory.
- **CPD3726**: `SAVACTWAIT` or `SAVACTMSGQ` parameters not valid with `SAVACT(*NO)`.

- **CPF3700 series**: Errors propagated during the QANESAVA save session (e.g., stream file locked in exclusive use).

RSTLIBIFS (Restore Library from IFS)

The **RSTLIBIFS** command restores an entire library (or part of it) **by reading the save stream directly from a stream file (STMF) residing in the IFS**, without needing to manually create an intermediate Save File (*SAVF). The command decodes data blocks using the **QANERSTA** system API.

Restrictions and Prerequisites

- **Pro Edition:** Command reserved for the full version of the BKPIFS suite (requires a valid activation key in the BKPIFSAUT data area).
- **Special Authorities:** Requires ***SAVSYS** special authority, or specific ***READ** and ***EXECUTE** authorities on the source IFS directory along with object authority for the target library.
- **Thread Safety:** The command is conditionally thread-safe. It must not be run in multi-threaded environments if the IFS stream file is locked exclusively by another process.

Command Parameters

Parameter (Keyword)	Description and Accepted Values	Type / Default
SAVLIB	**Saved Library.** Specifies the original name of the library contained in the backup file within the IFS.	*NAME *Required*
STMF	**IFS File.** The absolute path of the stream file (e.g., /backup/lib/mylib.bak) from which to read the restore stream.	*PATH *Required*
OPTION	**Restore Option.** Defines how objects are handled based on their existence or status. • *ALL: Restores all objects. • *NEW: Restores only objects not currently on the system. • *OLD: Restores only existing objects. • *FREE: Restores only objects whose storage has been freed.	Predefined options Default: ***ALL**
MBROPT	**Database Member Option.** Specifies how physical database file members are handled. • *MATCH: Members must match. • *ALL: Restores all members. • *NEW: Only new members. • *OLD: Only existing members.	Predefined options Default: ***MATCH**
ALWOBJDIF	**Allow Object Differences.** Handles discrepancies (e.g., different owner IDs or file level IDs). • *NONE: No differences allowed. • *ALL: Allows all differences. Cumulative multiple values (up to 4): *AUTL, *FILELVL, *OWNER, *PGP.	Single value or list Default: ***NONE**
RSTLIB	**Restore to Library.** Specifies where to restore the objects. • *SAVLIB: Restores into the original saved library. • *library-name*: Allows redirection (Restore to a different library).	Library name Default: ***SAVLIB**

OUTPUT	**Output.** Specifies whether to generate a listing of restored objects. • `*NONE`: No printed output. • `*PRINT`: Generates a spooled report (`QPSAVOBJ`).	Predefined options Default: **`*NONE`**

Practical Usage Examples

Example 1: Standard restore (overwrite)

Restores the `PROD_LIB` library to its original location by reading the backup file located in the `/backups/giornalieri/` IFS folder.

```
RSTLIBIFS SAVLIB(PROD_LIB) STMF('/backups/giornalieri/prod_lib.bak')
     OPTION(*ALL)
```

Example 2: Redirection and difference handling

Restores the contents of the `PROD_LIB` backup into a test library named `TEST_LIB`, ignoring object owner security differences and generating a printed report.

```
RSTLIBIFS SAVLIB(PROD_LIB)
     STMF('/backups/giornalieri/prod_lib.bak')
     RSTLIB(TEST_LIB)
     ALWOBJDIF(*OWNER *AUTL)
     OUTPUT(*PRINT)
```

Error Handling and Escape Messages

The command traps native restore system errors and IFS access issues, returning the following `*ESCAPE` messages:

- **`CPF9801`**: Object not found (the library or file does not exist).
- **`CPF9802`**: Authority failure for the object or IFS directory.
- **`CPF9810`**: Target library not found on the system.
- **`CPF9820`**: Security violation using the current IFS directory.
- **`CPF3800`** / **`CPF3700`** **series (indirect)**: General errors during the restore session driven by the `QANERSTA` API (e.g., corrupted or truncated file).

RSTOBJIFS (Restore Object from IFS)

The **RSTOBJIFS** command restores a single object or a group of objects within a single library onto the system, **reading the data directly from a stream file (STMF) residing in the IFS**. The command restores both the description and the contents of each specified object.
Special Cross-Restore Capability: The command is designed to accept both stream files specifically generated via SAVOBJIFS and full backup stream files generated via SAVLIBIFS. This allows the operator to extract and recover individual objects from a complete library backup without restoring the entire library.

Restrictions and Prerequisites

- **Pro Edition:** Command reserved for the full version of the BKPIFS suite (requires a valid activation key in the BKPIFSAUT data area).
- **System Authorities:** Requires ***SAVSYS** special authority.
- **Alternative Authorities:** Without *SAVSYS, the user must possess:
- Add (***ADD**) and execute (***EXECUTE**) authorities on the target library.
- Object existence authority (***OBJEXIST**) or ownership, if the object already exists on the system.
- *OBJEXIST and use (***USE**) authorities for message queues.
- *ADD authority for each user profile that owns the restored objects if those profiles do not yet exist on the system.
- **Program Limitations:** If restoring an OPM program, it must not be running during the restore, or unpredictable application behavior may occur.
- **User Domain Objects:**
  Objects of type *USRSPC, *USRIDX, or *USRQ restored into libraries that do not allow user domain objects (QALWUSRDMN) will automatically be converted into system domain objects.

Command Parameters

Parameter (Keyword)	Description and Accepted Values	Type / Default
OBJECT	**Objects.** Specifies the names of the objects to restore. • *ALL: Restores all saved objects matching the specified filters. • *object-name*: Specific name of a single object. • *generic-name**: Allows using an asterisk for groups of objects (e.g., PAG*). Supports up to 300 repetitions.	Single value or list *Required*
SAVLIB	**Saved Library.** Specifies the original library name that contained the objects at save time. Must match the exact name used in the backup.	*NAME *Required*
OBJTYPE	**Object Types.** Specifies the system object types to restore (e.g., *FILE, *PGM, *CMD). • *ALL: Restores all saved object types matching the OBJECT filter. Supports up to 300 repetitions.	Single value or list Default: ***ALL**

STMF	**Stream File.** The absolute path of the IFS file (generated by `SAVOBJIFS` or `SAVLIBIFS`) from which to extract restore data.	`*PATH` *Required*
OPTION	**Restore Option.** Controls behavior based on object existence. • `*ALL`: Replaces existing versions and adds missing objects. • `*NEW`: Restores only objects that do not exist on the system. • `*OLD`: Replaces only objects that already exist on the system. • `*FREE`: Restores only objects whose space has been freed on the system.	Predefined options Default: **`*ALL`**
MBROPT	**Database Member Option.** Specifies how to restore members of existing physical files. • `*MATCH`: Saved member list must match the system list exactly (not valid with `ALWOBJDIF(*ALL)`). • `*ALL`: Restores all saved members. • `*NEW`: Only members not known to the system. • `*OLD`: Only members already known to the system.	Predefined options Default: **`*MATCH`**
ALWOBJDIF	**Allow Object Differences.** Defines whether to ignore discrepancies between saved objects and system objects (requires `*ALLOBJ` special authority). • `*NONE`: No differences allowed. • `*ALL`: Allows all differences (renames files and members with different level IDs before restoring). Cumulative multiple values (up to 4): `*AUTL` (authorization lists), `*FILELVL` (file/member level ID), `*OWNER` (owner), `*PGP` (primary group).	Single value or list Default: **`*NONE`**
RSTLIB	**Restore to Library.** Specifies the target library for the objects. • `*SAVLIB`: Restores into the original saved library. • *library-name*: Allows redirecting objects into a different library.	Library name Default: **`*SAVLIB`**
OUTPUT	**Output.** Specifies whether to generate a job spool listing with detailed information (restored, not restored, excluded objects, and security info). Values: `*NONE`, `*PRINT`.	Predefined options Default: **`*NONE`**

System Behavior and Security

- **Ownership Management:** If the owner user profile recorded in the backup does not exist on the target restore system, the object is assigned to the default system profile **`QDFTOWN`**.
- **Audit Retention:** If an existing object is overwritten, it retains its current system `AUDIT` settings. If restored as a new object, it inherits the `AUDIT` settings held at save time.

Practical Usage Examples

Example 1: Targeted extraction and restore from a full backup (Cross-Restore)

Restores the `CLIENTI00F` database file into its original `PROD_DATA` library, extracting it directly from the full library backup file (`PROD_DATA_completo.bak`) residing in the IFS.

```
RSTOBJIFS OBJECT(CLIENTI00F) SAVLIB(PROD_DATA) OBJTYPE(*FILE)
     STMF('/backups/PROD_DATA_completo.bak')
```

Example 2: Scheduled restore into a Test library with owner tolerance

Restores all programs (`*PGM` objects) starting with "WEB" into the `TEST_LIB` library (originally saved from `PROD_LIB`), allowing restore even if the owner on the current system differs from the backup.

```
RSTOBJIFS OBJECT(WEB*)
     SAVLIB(PROD_LIB)
     OBJTYPE(*PGM)
     STMF('/backups/oggetti/web_tools.bak')
     RSTLIB(TEST_LIB)
     ALWOBJDIF(*OWNER)
```

Error Handling and Escape Messages

The command traps exceptions generated by the OS restore subsystem and IFS communication issues, returning the following `*ESCAPE` messages:

- **`CPF9801`**: Object not found (the target library or specified IFS stream file does not exist).
- **`CPF9802`**: Insufficient authority (user lacks permissions to access the IFS directory or target library).
- **`CPF3773`**: Objects restored or not restored (sent when the restore completes, but some objects specified in the list or generic filter failed or were excluded).
- **`CPF3800`** / **`CPF3700`** **series**: Critical session errors propagated from the `QANERSTA` API (e.g., corrupted stream file, premature data stream interruption, or exclusive lock held by another job).
- **`CPF9820`**: Security violation using the current IFS directory.

RSTLNKIFS (Restore Links from IFS)

The **RSTLNKIFS** command restores stream files, directories, and entire folder hierarchies within the Integrated File System (IFS), **reading data directly from a backup stream file (STMF)** previously generated with SAVLNKIFS. Leveraging the QANERSTA system API, it avoids creating intermediate Save Files (*SAVF) on disk.

Restrictions and Prerequisites

- **Pro Edition:** Command reserved for the full version of the BKPIFS suite (requires a valid activation key in the BKPIFSAUT data area).
- **System Authorities:** Requires ***SAVSYS** special authority, or specific write (***W**) and add (***ADD**) authorities on the target directory and execute/traverse authority (***X**) on the entire parent path.

Command Parameters

Parameter (Keyword)	Description and Accepted Values	Type / Default
OBJ	**Object.** Specifies the absolute path of the IFS object or directory to restore (up to 30 different paths). • *path*: Full IFS path saved in the backup file (e.g., '/Company/Documents').	IFS Path (*PNAME) *Required*
STMF	**IFS File.** The absolute path and name of the stream file from which to read the restore stream (e.g., '/QNAP_BACKUP/ifs_doc.bak').	IFS Path (*PNAME) *Required*
SUBTREE	**Directory Subtree.** Specifies how subdirectories are handled during the restore. • *ALL: Restores the entire directory tree (directories, subfolders, and all files). • *DIR: Restores only objects in the main directory without descending into subfolders. • *NONE: Restores only the directory object itself. • *OBJ / *STG: Advanced options for restoring only objects or storage.	Predefined options Default: ***ALL**
CRTPRNDIR	**Create Parent Directories.** Indicates whether the system should automatically create parent directories in the path if they do not yet exist on the target system. • *NO: If the parent directory is missing, the restore fails. • *YES: Automatically creates the missing parent directory tree.	Predefined options Default: ***NO**

Practical Usage Examples

Example 1: Full restore of an application folder

Restores the entire /WebApps/Portale directory and all its subdirectories by reading data from the portale.bak stream file on the NAS.

```
RSTLNKIFS OBJ('/WebApps/Portale')
     STMF('/QNAP_BACKUP/ifs/portale.bak')
     SUBTREE(*ALL)
```

Example 2: Restore to a new path creating parent folders

Restores the `/FattureElettroniche/2026` folder, automatically creating any missing intermediate directories on the system.

```
RSTLNKIFS OBJ('/FattureElettroniche/2026')
     STMF('/QNAP_BACKUP/ifs/documenti_fe.bak')
     SUBTREE(*ALL)
     CRTPRNDIR(*YES)
```

Error Handling and Escape Messages

- **CPFA0A9**: Backup stream file (`STMF`) or parent path not found.
- **CPFA09C**: Insufficient authority on the target IFS directory or backup file.
- **CPF3800** / **CPF3700** **series**: Errors propagated during the `QANERSTA` restore session (e.g., objects locked in exclusive use or corrupted stream file).