

## iSYNC File Data Replication

*Version 4.2.0.5*

## Table of Contents

<b>1.</b>	<b>Introduction .....</b>	<b>4</b>
1.1	Unique index based synchronisation algorithm .....	4
1.2	RRN-based replication algorithm .....	5
1.3	Unique index based replication for logical files .....	6
1.4	Journal-based replication option .....	6
1.5	File data archival .....	7
<b>2.</b>	<b>Principles of Operation .....</b>	<b>8</b>
2.1	Copy-based options .....	8
2.1.1	<i>*RRN option .....</i>	<i>11</i>
2.1.2	<i>*LF option .....</i>	<i>11</i>
2.2	Journal-based option .....	11
2.3	Data archival .....	12
2.4	Replication sets .....	12
<b>3.</b>	<b>Configuration and Monitoring .....</b>	<b>13</b>
3.1	Configuration and Operation .....	13
3.2	Control process .....	18
3.3	Communications configurations .....	18
3.3.1	<i>Intra-system data replication .....</i>	<i>18</i>
3.3.2	<i>Cross-partition data replication .....</i>	<i>19</i>
3.3.3	<i>ObjectConnect related settings .....</i>	<i>21</i>
3.4	Product APIs .....	21
3.4.1	<i>CRCESTI .....</i>	<i>21</i>
3.4.2	<i>CRCESTS .....</i>	<i>22</i>
3.4.3	<i>CRCEICB .....</i>	<i>22</i>
3.4.4	<i>CRCECKP .....</i>	<i>22</i>
3.5	Performance and Monitoring .....	23
3.5.1	<i>iSYNC dashboard .....</i>	<i>23</i>
3.5.2	<i>Multistreaming .....</i>	<i>25</i>
3.6	Resetting and recovering replication processes .....	26
3.6.1	<i>Introduction .....</i>	<i>26</i>
3.6.2	<i>Reset for journal-based replication .....</i>	<i>26</i>
3.6.3	<i>Recovery scenarios for journal-based replication .....</i>	<i>27</i>
3.6.4	<i>Recovery scenarios for the archival processes .....</i>	<i>28</i>
<b>4.</b>	<b>System requirements .....</b>	<b>29</b>

## **Trademarks**

Any trademarks and product or brand names referenced in this document are the property of their respective owners.

## 1. Introduction

iSYNC File Data Replication is an option of the iSTREAM Licensed Product. It was previously known as iSTREAM File Replication. This option has been designed as an easy-to-use file synchronisation mechanism. Later, more sophisticated functions, including logical file data replication, remote replication and data archival have been included in the product.

iSYNC offers two alternative synchronisation mechanisms, basic, making use of straightforward file comparison and update, and advanced relying on object journaling. The former can only be used for intra-system file data synchronisation, while the latter allows for both intra-system and cross-partition replication.

Throughout the manual the terms "source file" and "target file" are used to describe the file being replicated and the target file of the replication process.

Depending on the needs, the following data replication algorithms are available to select from for every given file (or table).

### 1.1 *Unique index based synchronisation algorithm*

This algorithm is based on periodic scanning of both the source and the target physical files (tables) and replicating the data changes accumulated since the last scan to the target file (table). The algorithm requires both the source and the target files to have unique key access paths (indexes or logical files). The access paths do not have to be explicitly defined as unique - it is sufficient for them to contain unique keys that would unambiguously define the related records (rows).

While the data from the source file (table) is replicated to the target, corresponding records may end up having different Relative Record Numbers (RRNs).

This algorithm has a number of limitations and prerequisites. The most important of them are the following:

- Replication is not instantaneous. At the same time, synchronisation of files using this algorithm is usually faster than copying data across with CPYF or SAVxxx and RSTxxx commands.
- Replication of multi-member files is not supported.
- Replication of multi-format files is not supported.
- Each file subject to replication must have a logical file defining a unique primary key. This file has to share the format with the physical file.
- Replication of tables containing BLOB and CLOB columns is not supported.

During the replication process the target files are locked by synchronisation processes. The level of the locking can be configured.

**Warning:** if values of the key fields of a source file record are changed during an update operation, the replication processor adds the updated version of the record to the target file and deletes the old one. Updates of the values of the key fields on a large scale may cause performance degradation of the replication process.

Each of the files is replicated by an independent process, so referential integrity settings may render iSYNC unsuitable for some configurations.

Existing referential constraints may cause replication issues when using this algorithm.

---

- iSTREAM Options 1 (Flash Operations) and 7 (iSYNC File Data Replication) must be installed and licensed in order to enable the Replication utility.
- iSTREAM Option 4 (Multistreaming Toolkit) must be installed and licensed, if multistreamed synchronisation is requested for any of the files.
- IBM i Object Connect feature of OS must be installed in the partition.

In spite of the above limitations, this file synchronisation algorithm offers multiple advantages. Firstly, file journaling is not required for the replication. Secondly and, perhaps, more importantly, this algorithm can synchronise files irrespectively of their initial contents.

## **1.2 RRN-based replication algorithm**

This algorithm is similar to the previous one, with the exception of unique access paths not being used. Each record of the source file (table) is replicated to the target file (table) using its RRN. As a result, not only the data in the files (tables) ends up being identical, but also the corresponding records (rows) always have the same RRNs.

Due to the nature of the algorithm, the source and the target files, however, may end up having different numbers of deleted records.

If larger numbers of records are added to the source file (table), this algorithm may not perform as well as the unique key based replication.

The target files are periodically locked by the synchronisation processes using \*EXCLRD locks; therefore, only read access to the target files is possible during the replication.

---

**Warning:** This algorithm is generally incompatible with any type of referential constraints and unique indexes defined for any of the existing access paths.

---

The algorithm has the following limitations:

- Replication of multi-member files is not supported.
- Replication of multi-format files is not supported.
- Replication of tables containing BLOB and CLOB columns is not supported.

### **1.3 Unique index based replication for logical files**

This is variation of the algorithm used for the key based replication of physical files (Section 1.1). It extends the functionality of the physical file related algorithm by allowing data replication from multiple physical files simultaneously. For that, join logical files can be used. The data is replicated to either a physical or a logical file.

This is the only option efficiently implementing multistreamed replication (see Section 2 for detail).

The algorithm has the following limitations:

- Replication of multi-member files is not supported.
- Replication of multi-format files is not supported.
- Replication of files referencing BLOB and CLOB columns of SQL tables is not supported.

### **1.4 Journal-based replication option**

This algorithm makes use of the traditional journal-based approach where source file record updates are picked up from a journal and replayed to the target file. The algorithm offers both intra-system and cross-partition replication options.

This algorithm also has a few limitations and requirements:

- Replication of multi-member files is not supported.
- Replication of multi-format files is not supported.
- Replication of tables containing BLOB and CLOB columns is not supported.
- iSTREAM Options 1 (Flash Operations) and 7 (File Replication) must be installed and licensed in order to enable iSYNC.
- IBM i Object Connect feature of OS must be installed in the partition(s).
- For the cross-partition replication option, communication between partitions is based on the SNA architecture making use of DDM files and APPC devices. (Cross-partition replication over IP can be made available as a custom PTF.)

Potentially, the journal-based algorithm can support replication of a larger number of files than the copy-based algorithms. In order to use the journal-based option, however, the source and the target files have to be initially synchronised.

### ***1.5 File data archival***

File data archival complements journal-based replication by transferring deleted records from the source file to a designated archive file on the target system.

## 2. Principles of Operation

### 2.1 Copy-based options

iSYNC File Data Replication utility is controlled by the contents of the CRCECFG file stored in the so called replication control library. Each record of CRCECFG contains replication parameters for a single file. The field structure of CRCECFG file is as follows.

- FLIB - name of the library containing the file to replicate (synchronise)
- FFIL - name of the file to replicate (synchronise)
- TLIB - target replication library
- TIND - name of the file in the target library defining a keyed access path (unique) to the physical file FFIL, or one of the control words, \*RRN, \*JOURNAL, or \*LF. If the value of the field is \*LF, the FFIL file in library TLIB must have a keyed access path (unique). This file must allow writes and updates.
- KEY1-KEY15 - names of the unique key fields in the TIND logical file (values are assigned to this field automatically). None of the fields in the list can be null-capable. The values are extracted from the definition of the TIND access path and assigned to these fields by the replication definition process. These fields are not used by the RRN or journal-based based replication algorithms.
- FRMT - name of the file format that must be shared between the file being defined for replication and the TIND logical file. Values are assigned to this field automatically.
- JOB - name of the job description used to submit the replication process for the given file. It is located using the library list of the job requesting the replication process to be started. If the field is left blank, the default job description is used (QBATCH).  
If multistreaming of the replication process is defined, the content of the field is treated as the job description pattern (see the section for JDPATTERN parameter of DFNSPTPRM command in the iSTREAM Generic Multistreaming Toolkit manual).
- STREAMS - number of streams the replicator process submits for each iteration of the synchronisation catch-up (extra system resource is used in this case). This feature is supported for the key-based replication and replication of logical files.



## iSYNC File Data Replication

- **INCRRN** - in some cases, the number of records at the beginning of the file represents the data that never changes. If this is the case, the value of this field can be set to the maximum RRN number in the related data section. Synchronisation processors then ignore the records in the static section. The never changing sections of the source and the target files must be identical at the time the replication process is submitted. By default, the value in the field is 0.  
For \*JOURNAL type of replication this value is used as the size of the ODP buffer used for the replication "insert" operations. By default (if the value of the field is 0), the size of the buffer is assumed to be 1000 records. Setting the value of this field to 1 effectively disables replication data buffering. This can be useful, if records belonging to different files must be replicated in the same sequence.  
The parameter is not supported for \*LF type of replication. The value of the field can be modified using product APIs.
- **EGRP** - this field is reserved for the exclusive use by the journal-based replication algorithm. Values are assigned to this field automatically or using product APIs.
- **LOCKREQ** - the field can be used to define the type of lock placed on the target file and index by the replicator service job. The values supported are \*SHRUPD and \*EXCLRD. By default, \*SHRUPD lock is requested. The field is added to CRCECFG files created by earlier modifications of iSTREAM automatically. The above locks are only requested in the case of intra-system replication.

iSYNC does not require all files being replicated to be stored in the same library. Target libraries for different files in a group can also be different. At the same time, each source file must be uniquely represented in the CRCECFG configuration.

A sample CRCECFG file is included in the iSTREAM Option 7 distribution. The file must be edited in order to configure the desirable replication. This can be done using the IBM i DFU utility. Alternatively, individual synchronisation definitions (records) can be added to the configuration file using ADDSYNDFN command. Once the replication parameters have been entered into CRCECFG file, DFNRPLUNT and CMPRPLUNT commands have to be executed in order to create replication work objects and compile the actual synchronisation programs. For each file being synchronised a program with a unique name is created in the configuration library.

---

**Warning:** If a new record has been added to an already existing CRCECFG configuration file using ADDSYNDFN command, DFNRPLUNT command has to be executed again in order to update the replication configuration.

---

When the replication process is requested using STRRPLUNT command, multiple jobs are submitted using the iSTREAM Asynchronous (Flash) Execution framework. One of the submitted jobs acts as a trigger releasing the synchronisation catch-up processes at regular intervals. The replicator process



## iSYNC File Data Replication

jobs remain active until the replication process is explicitly ended by the ENDRPLUNT command.

If the value of the STREAMS parameter for any of the files is greater than 1, the multistreaming feature of iSTREAM is implicitly invoked and configured for the given file. If option 4 of iSTREAM is not installed or the hot library is not configured for the unit, the compilation process ends with an error.

---

**Warning:** Multistreamed replication may perform poorly in the case of key-based physical file replication due to the OPNQRYF processor used for iSTREAM multistreaming implicitly changing the record arrival sequence for each of the streams. It is therefore only advised to use multistreaming when replicating logical files.

---

Multistreaming is automatically configured for the key-based physical and logical file based methods. The split is created over the first key field defined. Value ranges are calculated automatically on the basis of the data currently available in the data file. Each such breakdown should be verified before submitting the related replication process (more details can be found in the iSTREAM Generic Multistreaming Toolkit manual).

INCRRN parameter is ignored, if multistreaming is defined.

These automatically created multistreaming configurations can later be manually updated for higher efficiency.

---

**Warning:** If the library unit has been defined with no hot library, replication multistreaming is not compiled or activated, irrespectively of the values in the STREAMS field of the CRCECFG configuration file.

---

### 2.1.1 \*RRN option

This option can be used when a unique index for the file being replicated does not exist. It is similar to the index-based method, but should not be used when an object containing a keyed access path with \*UNIQUE property or \*UNIQUE referential constraint is defined for the file.

This algorithm can lock the target files for \*EXCLRD, so updating the target files while the replication process is running should not be attempted.

### 2.1.2 \*LF option

This option is used for replicating files with existing keyed access paths. Such paths must define unique indexes, although \*UNIQUE property of the access path is not a requirement. This kind of replication generally does not perform as well as replication of physical files, but offers two advantages over it:

- availability of an efficient multistreamed replication mechanism
- support of data replication based on join logical files, i.e. files including data from multiple physical files.

---

**Warning:** Care should be taken when L-replication is defined for objects created using DDL with large-scale numerics, e.g. of the type INTEGER, used as part of the index. The programs generated by iSYNC for replication may have to be manually tweaked, since automatic mapping of such fields to RPG structure subfields may cause unsigned numerics to be treated as signed.

---

## 2.2 Journal-based option

The basic copy-based option does not require journaling of the files being replicated but consumes CPU, disk and memory resource at each cycle. The latter remains true regardless of the number of actual file updates.

The journal-based option requires all source files in the replication set to be journaled to the unit journal with \*AFTER images and consumes system resource only when there are updates to process.

Configuration of the journal-based option is very similar to the one described above. The value of the TIND parameter must be set to the special value \*JOURNAL.

The values of the KEYxx, STREAMS and INCRRN parameters are ignored by the journal-based replication processes.

For the journal-based replication the source and the target files must contain the same data at the point in time when the replication is submitted. Alternatively, the replication process can be submitted later, but in that case *ISYNSTR* checkpoint (see the description of CRTCKP command in the *iSTREAM CL Command Transformer (CCT) Guide*) must be manually taken to help the replication process identify the starting entry in the journal. The actual name of

the checkpoint should follow the format *@untNT*, where the first character is the value of the system variable RPLSP (@ being the default), *unt* - the name of the iStream unit, N (0-9) - the number of the replication set in the unit (see section 2.4 for more detail), and T - the type of the checkpoint, "S" corresponding to the replication starting point.

By default, a separate processor is submitted for each file being replicated, exactly as is the case for the copy-based replication. However, it is possible to explicitly define the number of streams or groups (implemented as separate service jobs) submitted to apply record updates to the target files. This number is defined using CMPRPLUNT command. In many cases limiting the number of replication process jobs helps save system resource.

EGRP field values in the CRCECFG file are set by the CMPRPLUNT command processor to the sequential number of the stream (group) the given file replication process is allocated to. It is not recommended to change the values in this field using unsupported external interfaces.

If the journal-based replication option is configured for cross-partition replication, special care has to be taken of the communications configuration. See section 3.3 for the detailed description of the recommended communication configuration options.

## **2.3 Data archival**

Data archival is configured in exactly the same way as journal-based replication above, the only difference between the two being the way journal entries are processed. The data archival option ignores all journal entries except for those having "DL" and "DR" types. When such an entry is encountered, it is treated as a new record insert, and the corresponding data is added to the target file.

## **2.4 Replication sets**

In certain cases different files related to an iSTREAM unit may have different replication requirements. Not only the actual data synchronisation algorithms may be different, but also their synchronisation may be turned on and off at different points in time. The mechanism that can be used to make replication for different groups of files in a unit completely independent is call replication sets.

Up to 10 such sets can be defined for each iSTREAM unit. A set has its own iSYNC control library and CRCECFG configuration. All iSYNC line commands are scoped to just one replication set; thus, replication for different groups of files of a unit can be started, stopped, monitored and recovered independently.

The sets have names \*PRIM (primary set) and \*SET1-\*SET9. There is no difference in their functionality, but replication configurations from earlier releases of iSYNC are used to create the primary unit set when iSTREAM LP is upgraded to a release supporting replication sets.

### 3. Configuration and Monitoring

#### 3.1 Configuration and Operation

Most of the File Replication configuration commands are available from FILRPL menu:

```
FILRPL                                iSTREAM file replication menu                                System:  LS057

Select one of the following:

    1. Start replication for unit
    2. Prestart replication for unit

    4. End replication for unit
    5. Display file synchronisation processes
    6. Define replication for unit
    7. Compile replication for unit

    11. Start file synchronisation
    12. Start group synchronisation
    13. Reset file synchronisation
    14. Reset group synchronisation

    16. Display requests in queue

More...

Selection or command
===>

F3=Exit  F4=Prompt  F9=Retrieve  F12=Cancel  F16=System Main menu
```

In order to configure replication, a library for storing replication subsystem objects has to be created. This library has to then be defined as the replication control library for a specific replication set of the unit by executing DFNRPLUNT command. Optionally, this command can create a copy of CRCECFG configuration file from a template and invokes DFU utility allowing the user to define the list of files to be replicated or archived. (CRCECFG file can be pre-created and stored in the replication control library of a set for further editing and processing).

In order to add data to the CRCECFG configuration file, IBM DFU utility can be used (optionally invoked by DFNRPLUNT command processor). Alternatively, ADDSYNDFN command could be invoked manually or programmatically to add a record to the CRCECFG configuration file. ADDSYNDFN command creates both the iSYNC control library and the configuration file, if they do not exist.

**ADDSYNDFN** command adds data to the following CRCECFG file record fields:

- FLIB - name of the library containing the file to replicate (synchronise)
- FFIL - name of the file to replicate (synchronise)
- TLIB - target replication library
- TIND - name of the logical file in the target library defining a unique record key, or one of the control words, \*RRN, \*JOURNAL, or \*LF.

## iSYNC File Data Replication

- **JOBID** - name of the job description used to submit the replication process for the given file.
- **STREAMS** - number of streams the replicator process submits for each iteration of the synchronisation catch-up (extra system resource is used in this case).
- **INCRRN** - maximum RRN number in the "archive", i.e. never changing data section.
- **LOCKREQ** - the type of lock placed on the target file and index by the replicator service job.

It is important to allow **DFNRPLUNT** command complete before proceeding to the replication compilation step.

It is not advised to store any user objects in the control library, as these objects may be deleted by iSYNC.

**DFNRPLUNT** command has the following parameters:

- Library unit name
- Replication set identifier
- Name of the replication control library
- Processing mode
- Archival filter
- Qualified name of the template (model) DDM file

The library unit can be any unit with or without libraries defined. Flash operations must be enabled for the unit using **ASYEXE(\*INDEP)** parameter of the **STRISTMOD** command. The value of **ASYDDM** parameter must be set to **\*NONE**. **WLC** group for the replication processes can be defined using **WLCASY** parameter of the **STRISTMOD** command. This group, if defined, is shared between the general flash service and replication processes. If journal-based replication is used for at least some files, the unit control library and unit journal must be defined.

The processing mode parameter can have the value **\*SYNC** or **\*ARC**. **\*SYNC** stands for replication, **\*ARC** - for archival. It follows that archival definitions cannot share a replication set with any data replication definitions.

Archival filter is an optional parameter that could be used to define the program name on each journal entry that would be checked to determine whether the record being deleted has to be archived. Records deleted by any other program will not be considered for archival.

The model DDM file is used to define a communications channel between partitions in the case of cross-partition data replication. The parameters of the model DDM file are further discussed in section 3.3.2.

The source and target libraries involved in the replication do not have to be defined as protected, control or hot libraries of the **iSTREAM** library unit used. The files replicated using the journal-based option must be journaled with **\*AFTER** or **\*BOTH** images to the unit journal.

DFNRPLUNT command adds the name of the library provided to the configuration of the iSTREAM library unit. The replication set and its configuration library are thus tied to the unit and remain part of the unit configuration until the related parameter value is either reset (set to \*NONE) or changed by another subsequent DFNRPLUNT command for the same library unit.

Once DFNRPLUNT command has been executed, CMPRPLUNT configuration command must also be run to generate and compile the replication code and other service objects. Compilation can only be submitted when all the source files defined for replication have been created in the related libraries.

On entry to the CRCECFG file, the value of FFIL field can be either a name of a file or a generic name, e.g. *abc\**, standing for all files with the names starting from "abc" string, or \*ALL - all files in the source library. Such values are expanded by the DFNRPLUNT processor. Only one \*ALL file definition can be entered for a configuration at a time. More \*ALL definitions can be added later.

DFNRPLUNT command adds the replication control library for the set selected to the library list of the current job.

When DFNRPLSET command is executed with parameter ILIB(\*NONE) replication for the set is disabled and the related control library, if it has been previously defined, is cleared. CRCECFG file, if exists, is not deleted.

The **CMPRPLUNT** processor compiles necessary synchronisation programs into the replication control library. Replication can only be defined for up to 2048 files.

CMPRPLUNT command has the following set of parameters:

- Library unit name
- Replication set identifier
- File name (\*NONE, name, or \*ALL)
- Batch mode (\*YES or \*NO)
- Compilation option (\*REPLACE or \*ADD)
- Number of streams (groups) for journal processing (number, \*NUMFILES, or \*SAME)

The command allows to selectively request the type of the compilation (parameter FILE). If its value is set to \*ALL, replication objects are created for all files defined in the CRCECFG configuration. If a new file is added to an already compiled configuration, service objects have to only be compiled for this file. Setting the value of the FILE parameter to \*NONE suppresses compilation of the file-related objects and can be used when there is a need to only redefine the number of journal processing streams (STREAMS parameter).

Execution of the compilation can optionally be submitted to batch - especially if the entire configuration requires recompilation.

The STREAMS parameter defines the number of journal processor jobs submitted by iSTREAM for journal-based replication or archival. Special value \*NUMFILES requests submission of a separate processor job for each of the files in the configuration. This is the default for a new configuration.

## iSYNC File Data Replication

CMRPLUNT command adds the replication control library for the set selected to the library list of the current job.

To start the actual replication process, **STRRPLUNT** command has to be executed. It has the following parameters:

- Library unit name
- Replication set identifier
- Synchronisation interval
- Preset checkpoint (\*YES/\*NO/\*PRV)

STRRPLUNT processor submits multiple file synchronisation jobs and a control process performing time management using an internal STRFS command. This command has to be added to the list of commands enabled for flash processing. The related ENAFLEXE command is automatically executed during the installation of Option 7 of iSTREAM (iSYNC).

Each of the synchronisation jobs has the name of the replication control library automatically added to its library list.

The time management process periodically triggers the catch-up activity in the submitted jobs that synchronises the files in the primary and target libraries using copy-based algorithms (if configured). The interval between subsequent synchronisation cycles is defined in seconds. 600-900 seconds is usually a good starting value that helps keeping the differences between the libraries to a minimum.

The journal-based replication mechanism does not use synchronisation cycles, as it implements continuous replication. However, the cycles are still used to run a periodic health-check of the replication process.

Journal-based replication requires a journal checkpoint to use as the replication starting point. By default, the replication process starts from the end of the journal. If, however, the source and the target files have been synchronised at a point in time in the past and a checkpoint (see section 1.4) with the name @*unt*NS (where *unt* is the name of the library unit and N - the number of the replication set) created to point at the related journal entry using iSTREAM CRTCKP command, specifying \*YES as the value of the PRESET parameter causes the replication process to replay all the journal entries starting from that checkpoint.

An example of this scenario can be illustrated by the following script.

1. Define *iSYNSTR* checkpoint
2. Submit *savrst-while-active* of the source files to the target library
3. Start updating source files
4. Wait for the *savrst* process to end
5. Start replication from the pre-created checkpoint

Ultimately, the source and the target files will be synchronised.

STRRPLUNT processor enters iSTREAM mode for the current job in the related library unit.



## iSYNC File Data Replication

Sometimes SAVRSTxxx command may be executed at the time when data is being added to application files. In this case, using the previously defined checkpoint may not be correct due to records that could have been added to the database between the taking of the checkpoint and the save-while-active process initialisation. In such cases, the PRESET parameter can be used to request an adjustment to the replication start entry. If the code, type and file object subfields of the PRESET parameter are defined, the replicator attempts to find the related entry searching the journal in the ascending sequence from the @untNS checkpoint-related entry. The entry, if found, is then used as the starting point of the replication process. While any journal file-related entry could be used for the above adjustment, in the case of SAVRSTxxx-while-active a possible combination of the journal code and type to be used is F-SS.

If the specified entry is not found in the journal, the adjustment function is not performed.

To define a replication start point **CRCECKP** API should be used.

This API verifies iSYNC internal structure consistency before creating the actual checkpoint.

If the replication process is ended using **ENDRPLUNT** command (see below), it can be restarted from the last journal entries processed by each of the replication service agents respectively. For that, PRESET(\*PRV) parameter must be used.

When it is time to end replication, input to the source files should be suspended and ENDRPLUNT command executed. This command communicates with the background processors allowing them in the case of copy-based synchronisation to run the last (unscheduled) synchronisation cycle. Once the cycle is over, the replication controller job ends and so do all the synchronisation jobs.

The process of ending journal-based synchronisation jobs is different. ENDRPLUNT command processor creates a *ISYNEND* checkpoint writing a special request entry to the journal, and synchronisation jobs complete only when they reach this entry.

If the replication process has to be ended immediately, e.g. without completing the last synchronisation cycle, ENDRPLUNT with TYPE(\*IMMED) parameter can be used. In that case, all the background jobs involved in the replication are ended immediately and ISYNEND checkpoint is not created. There is no guarantee that the primary and target files are synchronised, if replication is ended using this method. The active file synchronisation requests remain pending. They can be removed using option 4 (or PF4 key) from the **DSPFILSYN** command display.

DSPFILSYN processor enters iSTREAM mode for the current job in the related library unit.

If some of the replication agent jobs fail before ENDRPLUNT with TYPE(\*CNTRLD) parameter is executed, they can be individually restarted using STRFILSYN or STRGRPSYN commands. STRRPLUNT command with PRESET(\*PRV) parameter should not be used in this case.

**WAITRPLRQS** command can be used to programmatically determine whether there are any active replication processes and, optionally, wait for their completion.

The replication configuration can be disabled by executing **DFNRPLUNT** command for the given set in a unit by setting the value of the replication control library to **\*NONE**.

To entirely remove the replication configuration for the set, both hot library for the library unit and the library where **CRCECFG** file resides must also be deleted.

Time required to start data replication for a unit process largely depends on the number of the individual service processes configured. The more jobs to be submitted the longer it may take for the replication process to activate. The activation phase of the replication process can be accelerated by using the replication prestart facility (**PRSRPLUNT** command). This command submits all the service jobs required. The latter enter a dormant state until the following **STRRPLUNT** command is used. Once **PRSRPLUNT** command is executed, changing or recompiling the replication configuration for the related set is not allowed.

**PRSRPLUNT** processor enters **iSTREAM** mode for the current job in the related library unit.

The current status of the replication process (**INACTIVE**, **ACTIVE**, **STARTED**) is available from the **DSPFILSYN** command display.

A list of replication sets defined for the given unit can be generated using **LSTRPLSET** command.

### **3.2 Control process**

The control process (or job) performs multiple data replication-related system management functions. This job needs to be active until the entire replication process ends. If the control job ends abnormally, it can be restarted using **STRFILSYN** command with **\*CONTROL** parameter value.

If the control job is not active, the replication process cannot be ended normally. **ENDRPLUNT TYPE(\*IMMED)** command can always be used to both end all the synchronisation background processes and reset the replication process status to **INACTIVE**.

### **3.3 Communications configurations**

#### **3.3.1 Intra-system data replication**

In the case of intra-partition replication an intra-partition communications configuration must be created in order for the **RESET** function of the replication (implemented using IBM Object Connect commands) to be enabled.

## iSYNC File Data Replication

The intra-system SNA configuration required can be set up using the following commands:

```
CRTCTLAPPC CTLD(ISTREAMR) LINKTYPE(*LOCAL) ONLINE(*YES)
    TEXT('iSTREAM File Replication comms controller')
    CMNRCYIMT(2 5) MSGQ(*SYSVAL)
CRTDEVAPPC DEVD(IRFROM) RMTLOCNAME(IRTO) ONLINE(*YES)
    LCLLOCNAME(IRFROM) RMTNETID(*NETATR) CTL(ISTREAMR)
    MODE(QSOCCT) MSGQ(*CTLD) APPN(*NO) SNGSSN(*NO)
    SECURELOC(*NO) LOCADR(00)
CRTDEVAPPC DEVD(IRTO) RMTLOCNAME(IRFROM) ONLINE(*YES)
    LCLLOCNAME(IRTO) RMTNETID(*NETATR) CTL(ISTREAMR)
    MODE(QSOCCT) MSGQ(*CTLD) APPN(*NO) SNGSSN(*NO)
    SECURELOC(*NO) LOCADR(00)
```

### 3.3.2 Cross-partition data replication

The cross-partition data replication option requires two communications configurations to be created, one for the data replication function itself, and the other for the RESET function enablement.

Cross-partition data replication is implemented in iSYNC using DDM and ObjectConnect. Only SNA communications can be used by the current modification of iSYNC. For each file involved, a related DDM file is created to point to the related target file. Data synchronisation programs retrieve source file journal entries and update the target files using the related DDM file as a proxy.

The DDM files are created by the DFNRPLUNT processor from the model DDM file provided by the user. The name of the model DDM file must be the same as the name of the remote location used to communicate with the target partition. This remote location is also used by the RESET function to define the target of the Object Connect SAVRSTOBJ operation.

The following configuration can be used as an example.

#### Source partition

```
CRTDDMF FILE(REPCTRL/DDMT) 1
    RMTFILE(LIB1/FIL1)
    RMTLOCNAME(DDMT) 2
    MODE(QSOCCT) 3
    LVLCHK(*NO)

CRTCTLAPPC CTLD(HPRIP)
    LINKTYPE(*HPRIP)
    SWITCHED(*YES)
    RMTINTNETA('192.10.1.1') 4
    LCLINTNETA(*SYS)
    RMTNETID(*NETATR) 5
    RMTCPNAME(REMOTECP) 6
    DIALINIT(*IMMED)
    CPSSN(*NO)
    BEXROLE(*ENDNODE)
    USRDFN1(128)
    USRDFN2(128)
    USRDFN3(128)
```

## iSYNC File Data Replication

```

TEXT('iSTREAM File Replication Comms Controller')
DSCTMR(20 10)

CRTDEVAPPC DEVD(IRFROM) RMTLOCNAME(DDMT)          7
          ONLINE(*YES) LCLLOCNAME(IRFROM)         8
          RMTNETID(*NETATR)                       9
          CTL(HPRIIP)
          MODE(QSOCCT)                             10
          MSGQ(*CTLD) APPN(*NO) SNGSSN(*NO)
          SECURELOC(*NO) LOCADR(00)

```

### Remote partition

```

CRTCTLAPPC CTLD(HPRIIP)
          LINKTYPE(*HPRIIP)
          SWITCHED(*YES)
          RMTINTNETA('192.10.1.2') 11
          LCLINTNETA(*SYS)
          RMTNETID(*NETATR)         12
          RMTCPNAME(LOCALCP)       13
          DIALINIT(*IMMED)
          CPSSN(*NO)
          BEXROLE(*ENDNODE)
          USRDFN1(128)
          USRDFN2(128)
          USRDFN3(128)
          TEXT('iSTREAM File Replication Comms Controller')
          DSCTMR(20 10)

CRTDEVAPPC DEVD(DDMT)              14
          RMTLOCNAME(IRFROM)       15
          ONLINE(*YES)
          LCLLOCNAME(DDMT)         16
          RMTNETID(*NETATR)        17
          CTL(HPRIIP)
          MODE(QSOCCT)             18
          MSGQ(*CTLD) APPN(*NO) SNGSSN(*NO)
          SECURELOC(*NO) LOCADR(00)

ADDCMNE  SBSDB(QCMN)
          DEV(DDMT)                19
          DFTUSR(DBUPD)            20
          MODE(QSOCCT)             21

```

The values of the above parameters are as follows.

1. The name of the supplied model DDM file (has to be the same as 2).
2. The name of the remote location (has to be the same as 1).
3. Communications mode. Any mode could be used but QSOCCT is recommended, because a) it has sufficient number of source sessions and conversations configured (one session and conversation per synchronisation service job) and b) it can also be used by the RESET function, so no additional communications modes would be required.
4. TCP address of the remote partition.
5. Remote system's network ID (assumed to be the same as that of the local system)
6. Remote system's control point name - use DSPNETA command to retrieve.
7. Remote location name - same as 1 and 2.
8. Local location name - must be the same as 15.

## iSYNC File Data Replication

9. Remote network ID - same as 5.
10. Mode - same as 3.
11. Local system's IP address
12. Local system's network ID - same as 5 and 9.
13. Local system's control point name - use DSPNETA command to retrieve.
14. Remote device name - must be the same as 19.
15. Local device name - must be the same as 8.
16. Remote location name - must be the same as 1, 2 and 7.
17. Local network ID - same as 5 and 9.
18. Communications mode - same as 3 and 10.
19. Remote device name - same as 14.
20. Profile name on remote system for DDM communications. It could be defined as a no-password profile with authority sufficient for updating the required target files.
21. Mode - same as 3,10 and 18.

The above is just a sample configuration, but it illustrates the possible mapping of the runtime communications parameters.

### 3.3.3 *ObjectConnect related settings*

IBM ObjectConnect commands, e.g. SAVRSTLIB, may require a sophisticated security setting. For example, if new communication entries are added to QCMN subsystem for ObjectConnect, profiles specified on such entries must have sufficient authority in order to use the related IBM programs. Please refer to the IBM i system documentation for the details of the appropriate ObjectConnect security settings.

## 3.4 **Product APIs**

### 3.4.1 *CRCESTI*

This API can be used to update INCRRN value for a file defined in the CRCECFG configuration. The API is invoked by calling CRCESTI program in ISTSSYS library with the following parameters:

- FFIL - name of the file to replicate (synchronise) (CHAR 10). SQL wildcard characters (underscore and percent sign) can be used in order to update multiple file records at a time.
- FLIB - name of the library containing the file to replicate (synchronise) (CHAR 10). SQL wildcard characters (underscore and percent sign) can be used in order to update multiple file records at a time.
- INCRRN - the last RRN in the data section that is not to be updated (Decimal 15,5). Any negative value is treated as 0.
- Replication set identifier (CHAR 5).

The API can only be called from the iSTREAM mode environment for the related unit.

### 3.4.2 CRCESTS

This API can be used to update the number of the stream (group) the file is to be allocated to in the journal-based replication scenario. The API is invoked by calling CRCESTS program in ISTSSYS library with the following parameters:

- FFIL - name of the file to replicate (synchronise) (CHAR 10). SQL wildcard characters (underscore and percent sign) can be used in order to update multiple file records at a time.
- FLIB - name of the library containing the file to replicate (synchronise) (CHAR 10). SQL wildcard characters (underscore and percent sign) can be used in order to update multiple file records at a time.
- STREAM - the number of the stream (Decimal 15,5). Values from 0 to 99 are valid.
- Replication set identifier (CHAR 5).

The API can only be called from the iSTREAM mode environment for the related unit.

### 3.4.3 CRCEICB

This API can be used to form the name of checkpoints used by the replication control function. The API is invoked by calling CRCESTS program in ISTSSYS library with the following parameters:

#### Input

- UNIT - name of the unit (the value "\*" could be used to refer to the unit the current job has entered the iSTREAM mode for (CHAR 3).
- TYPE - type of the checkpoint ("STR", "END", "CYC", and "DUM" are supported). The *ISYNSTR* ("STR") checkpoint can be used in the STRRPLUNT PRESET(\*YES) scenario (CHAR 3).
- RPLSET - the name of the replication set (CHAR 5)

#### Output

- CKPT - checkpoint name (CHAR 7)

The API does not perform syntax checking of the input values.

### 3.4.4 CRCECKP

This API verifies internal iSYNC checkpoint structures for consistency and creates a replication start checkpoint. The API has two parameters:

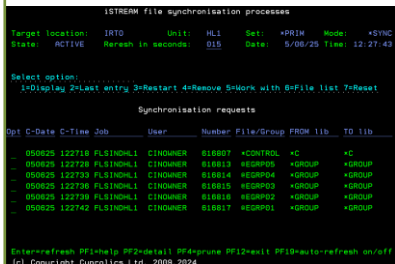
- UNIT (CHAR 3)
- RPLSET (CHAR 5)

### 3.5 Performance and Monitoring

#### 3.5.1 iSYNC dashboard

The iSTREAM File Replication makes use of the flash functionality packaged as option 1 of 7S77STR LP. DSPFLSRQS command can therefore be used to monitor the batch processes. A better option, however, is to use DSPFILSYN command for this purpose. It is similar to DSPFLSRQS but only allows to work with iSYNC jobs and provides access to additional information, e.g. time and date of the last synchronisation cycle for each of the file synchronisation processes. DSPFILSYN activates iSTREAM mode for the job, if it has not yet been activated.

The following screenshot illustrates the use of the DSPFILSYN command display for replication process monitoring:



```

iSTREAM file synchronisation processes
Target location: IRTD Unit: HLI Set: +PRIM Mode: +SYNC
State: ACTIVE Refresh in seconds: 015 Date: 5/06/25 Time: 12:27:43

Select option: ... entry 3=Restart 4=Remove 5=Work with 6=File list 7=Reset
1=Display 2=Exit entry 3=Restart 4=Remove 5=Work with 6=File list 7=Reset

Synchronisation requests
Opt C-Data C-Time Job User Number File/Group FROM lib. TO lib.
050625 122718 PLSINDML1 CINDHNER 818807 +CONTROL +C +C
050625 122728 PLSINDML1 CINDHNER 818813 +GROUP +GROUP
050625 122733 PLSINDML1 CINDHNER 818814 +GROUP +GROUP
050625 122736 PLSINDML1 CINDHNER 818815 +GROUP +GROUP
050625 122739 PLSINDML1 CINDHNER 818816 +GROUP +GROUP
050625 122742 PLSINDML1 CINDHNER 818817 +GROUP +GROUP

Enter/refresh PF1=help PF2=detail PF3=purge PF12=exit PF18=auto-refresh on/off
(c) Copyright Cyprolics Ltd. 2009, 2024
  
```

## iSYNC File Data Replication

This screen contains a list of file synchronisation processes with the last synchronisation date and time pair for each. The related processes can also be viewed using DSPFLSRQS command (see iSTREAM CCT Manual for details).

The "target location" field contains the name of the model DDM file in case of a cross-unit replication configuration, or, if the replication configuration defines intra-system synchronisation, the character string \*LOCAL.

Job, User and Number columns of the above screen contain the identifiers of the server job executing the related requests.

Unlike DSPFLSRQS, DSPFILSYN command enters iSTREAM mode for the parameter unit in the current job. The iSTREAM mode for the related unit must be activated in order to be able to reset any of the failed synchronisation processes.

Option 1 on the DSPFILSYN screen can be used to display the CL command used to start the process related to the entry. Option 2 displays information specific to journal-based replication. Status codes have the following meaning:

- 0 - Normal
- 1 - Unexpected journal entry
- 2 - Unexpected value in the first byte of the second parameter (>2) of the journal receive exit program
- 3 - Abnormal end of insert operation
- 4 - Record to be deleted or updated not found in target file
- 5 - DELETE operation failed
- 6 - UPDATE operation failed
- 7 - Unit configuration error (abnormal CRCVRT2 or CRCVRT6 response code)
- 8 - internal error processing CLRPFM condition (more info may be available in joblog)
- 9 - file incorrectly journaled for the required replication of j-type
- C - Replication cycled (not an error)
- D - System call with no entries (not an error)
- E - Synchronisation ended (not an error)
- I - Earlier error set to be ignored using CRCEIFE API
- N - New receiver attached (not an error)
- S - Replication started (not an error)

Option 3 ends the current job (if active) and resubmits the synchronisation process for the file. Option 4 can be used to remove the request related to a job that ended abnormally. Option 5 can be used to work with the job where the synchronisation process is running. Option 6 relates to journal based replication and displays a list of files replicated by the related journal processing stream. Option 7 copies the source the target location by using SAVRSTOBJ command and updates status data for the process.



### 3.5.2 Multistreaming

If a catch-up cycle for a file takes too long, copy-based processes for logical files can be multistreamed to improve their speed. The following definition can then be used as a template:

```
DFNSPTPRM PROGRAM(SCPF)  
    MASTERFILE((PRI/SC10LF *FIRST *NO *INP 1000 *ALLIO  
                1000 (&KEY1)) (TRG/SC10LF *FIRST *NO  
                *INP 1000 *ALLIO 1000 (&KEY1))) AUTOOVR(*YES)
```

According to the above definitions, the name of the synch program generated by iSTREAM as part of the configuration definition process is the same as the name of the file being replicated. The above definition has to be compiled (CMPSPDFN) into the hot library for the library unit and the number of streams defined, e.g.

```
ISTSSYS/INZSPTVAL UNIT(TRG)  
    PROGRAM(SC10LF)  
    STREAMS(10)
```

The multistreaming definitions are generated and processed automatically for the files having in the CRCECFG table STREAMS property set up. HOTLIB parameter in this case has to be explicitly defined for the library unit.

Although multistreaming can be defined for both physical and logical file replication, it is generally not recommended to configure it for the former due to potential performance degradation.

### 3.6 Resetting and recovering replication processes

#### 3.6.1 Introduction

A replication process for a file may fail for multiple reasons including corruption of the target file. If this is the case, the related entry on the flash process screen is displayed in red. It is usually possible to look at the job using option 5 and thus establish the reason for the failure. If the replication job ends without leaving a job log in the spool, it is recommended to explicitly define a job description for the file replication process (see 3.1 for more detail).

If the reason for the failure has been established, the replication process can be restarted with or without a reset.

Both types of restart require manual execution of **STRFILSYN** command (it can optionally be executed from the DSPFILSYN command screen using option 3, this type of request additionally ending the currently executing process, if active). The command has the following parameters:

- UNIT - iSTREAM unit used to control the replication process
- RPLSET - Replication set identifier
- FILE - qualified name of the file to be replicated
- RESET - \*YES or \*NO (not available for archival processes). Setting this parameter to \*YES causes an attempt by iSTREAM to use SAVRSTOBJ system command to replace the target file with a copy of the original file before running the next catch-up cycle.

The SNA configuration required can be set up using the commands described in Section 3.3.

An alternative way of resetting the replication relevant in the case of copy-based replication is to clear the contents of the target file. This, however, may cause the catch-up process to consume a lot of resource and take rather a long time to complete.

- INTERVAL - in the case of restarting the \*CONTROL process the new synchronisation interval has to be defined
- INCRRN - the maximum RRN of the archive (static section of the file at the top of it). If prompted, the value stored at the definition stage is used to populate the field. For journal-based replication the value of the field is used to define the size of the target file insert buffer used by the replication process.

STRFILSYN command enters the iSTREAM mode for the parameter unit in the current job.

Replication can also be reset for a single file using option 7 from the DSPFILSYN screen.

#### 3.6.2 Reset for journal-based replication

If a reset request is executed for a journal-based replication agent, the source file (or files in case of the agent being a stream replication agent) is copied using SAVRSTOBJ command executed in the while-active mode to the target library and

information about the last successfully processed journal entry for the agent is updated with one of the journal entry numbers of the F-SS type related to the SAVRSTOBJ operation.

**STRFILSYN** (or **STRGRPSYN** for stream processing agents) command restarts the agent immediately after the completion of the SAVRSTOBJ operation.

**STRGRPSYN** command performs a function similar to that **STRFILSYN** for a failed journal-based replication stream (group). Reset in the cause of restarting a group can only be requested if all source files allocated to the stream reside in the same library.

Replication of logical files cannot be reset.

**RESFILSYN** and **RESGRPSYN** commands can be used to reset previously failed replication processes for a single file or group of files respectively. These commands, like **STRFILSYN** and **STRGRPSYN**, do not check the status of the current replication processes. To avoid possible clashes, the related service jobs should be ended before any of the reset commands is executed.

**RESFILSYN** and **RESGRPSYN** commands cannot be used for archival processes.

### 3.6.3 Recovery scenarios for journal-based replication

The following recovery (and restart) scenarios are supported for journal-based replication.

a) Running replication is ended using **ENDRPLUNT** command with **TYPE(\*IMMED)** parameter. In this case no end checkpoint is added to the journal to indicate a request to end; all replication service jobs are ended with **\*IMMED** parameter. Each of the service jobs maintains the number of the most recent successfully processed journal entry.

To restart the replication the following options are available:

- Use **STRRPLUNT** command with **PRESET(\*PRV)** parameter. All processor jobs will be restarted from the journal entry following the latest processed.
- Use **STRRPLUNT** command with **PRESET(\*NO)** parameter. In this case, a new **@untSTR** checkpoint will be created at the end of the journal and replication will restart from that point.
- Use **STRRPLUNT** command with **PRESET(\*YES)** parameter. Replication will restart from the latest **@untSTR** checkpoint found in the journal.

b) Running replication is ended normally using **ENDRPLUNT** command with **TYPE(\*CNTRLD)** parameter. In this case an end checkpoint is added to the journal to indicate a request to end; all replication service jobs will end as soon as they reach the related journal entry.

To restart the replication the following options are available:

## iSYNC File Data Replication

- Use STRRPLUNT command with PRESET(\*PRV) parameter. All processor jobs will be restarted from the latest successfully processed journal entry, i.e. from the latest end request checkpoint.
- Use STRRPLUNT command with PRESET(\*NO) parameter. In this case a new @untSTR checkpoint will be created at the end of the journal and replication will restart from that point.
- Use STRRPLUNT command with PRESET(\*YES) parameter. Replication will restart from the latest @untSTR checkpoint found in the journal.

c) A failed replication request appears on the DSPFILSYN screen.

To restart the replication the following options are available:

- Reset the process using option 3 (or option 7) and restart it. In this case, the replication process will attempt to restart from the SAVRSTOBJ-related journal entry.

d) Replication is ended using ENDRPLUNT command with TYPE(\*CNTRLD) parameter, but the process times out or hangs due to failed agent processes. If the ENDRPLUNT command processor is still waiting, another screen can be used to display the second DSPFILSYN monitor screen. The command will show a list of pending replication requests.

To restart the replication the following straightforward option is possible:

- Service jobs for the pending requests have to be ended using ENDJOB command. Then, the requests themselves can be reset using option 7 from the DSPFILSYN screen. Once the processes have been reset, the related entries should be removed from the queue using option 4 (or PF4 key). If ENDRPLUNT command times out, it could be retried. After that, replication can be restarted using STRRPLUNT TYPE(\*PRV) command.

### 3.6.4 Recovery scenarios for the archival processes

If an archival process ends abnormally, it is necessary to find out the reason for the failure from the related joblog and information displayed by option 2 on the DPFILSYN dashboard screen for the failed process.

If no error code (status code from 1 to 8) is available, this means that the archival process failed without raising the error flag. In this case, it would be possible to restart the process to retry the failed operation.

If an error code is available, the failed process can be restarted. However, in order to do this the error flag raised by the failed process has to be dropped. This can be achieved using CRCEFIE API. The API has the following parameters:

- UNIT - name of the library unit
- RPLSET - replication set identifier
- FLIB - name of the library containing the source file
- FFIL - source file involved in the failed archival operation

#### **4. System requirements**

iSTREAM File Replication option can be used on IBM i 7.3+ platforms with the ILE RPG compiler and IBM i ObjectConnect feature of the Operating System installed and licensed.

LSTRPLSET command requires OS V7R4M0 or later to be installed.