Progress®



OpenEdge Database Replication Quick Start

OpenEdge®

Table of Contents

Main steps for OpenEdge database replication.....5
OpenEdge replication options.....6
OpenEdge replication prerequisites 7

Plan for OpenEdge Replication.....9
Prepare your production environment for replication.....9
Decide how you want replication to work.....13

Configure the property files.....21
Understand DSRUTIL STATUS vs -detail (or verbose) return codes.....23

Set Up OpenEdge Replication databases.....29
Configure the database service communication area size (-pica).....32

Start the replication.....33

1

Main steps for OpenEdge database replication

In today's world, organizations need to ensure that their data is protected against all kinds of loss such as hardware failures, user errors, database corruption, and natural or other disasters. One of the most important tasks you can perform is to backup and replicate your databases frequently. In this guide, first you will learn about what information goes into a database replication plan and then how to replicate with OpenEdge.

OpenEdge Replication is a process in which data is continuously replicated from a production database to one or two other databases using AI files. If the production database fails or is brought offline, then one of the other databases can temporarily become the production database to ensure availability of data access.

Replication eliminates the database as a single point of failure and minimizes business disruption.

In this guide, you will learn how to implement OpenEdge Replication.

Note: Progress also offers the OpenEdge Replication Plus option. OpenEdge Replication Plus allows you to run reporting (i.e., read-only) applications on the secondary replication databases, thus reducing the load on the primary, production database.

Guide objectives

When you complete this guide, you should be able to:

- Create a replication plan.
- Understand the differences between replication types.
- Implement replication with OpenEdge.

For details, see the following topics:

- OpenEdge replication options
- OpenEdge replication prerequisites

OpenEdge replication options

There are two options for implementing OpenEdge Replication—single-target replication and replication set.

Single-target replication consists of a Source database on the primary machine and a Target database on the secondary machine. The replication server that runs on the primary machine continuously sends database updates to the Target database via the replication agent running on the secondary machine. When the Source fails, the Target transitions to a new Source and replication stops. When a database is restored on the primary machine, it can be synchronized with the temporary Source on the secondary machine. Then, database activity can be failed back to the original configuration.

With a replication set, there are two Target databases. A replication set consists of a Source database on the primary machine and a Target database on each of two secondary machines. One of the Targets is configured as the primary Target, the other as the secondary Target. A replication server runs on the primary machine and a replication agent runs on each secondary machine. The replication server sends Source database updates to the replication agents, which in turn apply the updates to their Targets.

The advantage of a replication set is that if there is a loss of the Source or if the Source is brought down for maintenance, the primary Target becomes the new Source and continues replicating to the other Target. In this way, with a replication set, even if the Source database is down, replication continues.

The overall steps in the OpenEdge Replication process for both option are similar up to step six. At step six for a replication set, the procedure ends. The single-target replication requires additional steps.

1. During primary (normal) replication, the primary database has the role of source database and the secondary database has the role of target database. The Replication server exists on the source database and the Replication agent exists on the target database.
2. If there is a failure on the machine hosting the primary database, the Replication agent on the target database loses communication contact with the source database's Replication server.
3. The Replication agent on the target database enters pretransition.
4. Transition then occurs (automatically or manually, depending on how you set the transition properties), making the secondary database (formerly a target) a source database.
5. All database activity is moved to the secondary database, which is now functioning as a source database. The secondary database now becomes the production database.
6. For the Replication Set option the second target transitions to be an agent of the new secondary database.

For single-target replication after the machine hosting the primary database is fixed, the replication process continues as follows:

1. The primary database must be restored from a backup of the secondary database and then enabled as a target database before secondary replication can begin.
2. The new target can receive replicated data from the current source.
3. At a convenient time, you can perform the failback process to return the databases to the roles originally established, with the primary database as the source and the secondary database as the target. You can do this by transitioning the secondary database using the failover command.

OpenEdge replication prerequisites

Before you can begin replicating with OpenEdge, you must complete the following prerequisites.

- Understand source and target architecture requirements
- Identify a database that you want to replicate
- Create a database backup plan
- Back up your source database

Source and target architecture requirements

Before OpenEdge Replication starts, the source and target databases are automatically checked to ensure that the databases are identical in the following ways:

- The logical structure—but not necessarily the physical structure—of the databases. (All user-defined areas must be identical except the AI areas.)
- The versions of the databases.
- The supported platform.
- The database block sizes.
- If large file support is enabled, it must be enabled on both databases.
- Before-Image (BI) block sizes.

In addition, the values on the target database must be greater than or equal to the values on the source for the following database startup parameters:

- Lock Table Entries (`-L`)
- Number of Users (`-n`)
- Maximum JTA Transactions (`-Maxxids`)
- The sum of Number of Users (`-n`) and Maximum Servers (-Mn)

If the values for the target are less than the source, the agent will not start. Ideally the values of `-L`, `-n`, `-Maxsids`, and `-Mn` should be equal on for the source and target databases.

For added protection against data loss, locate the target database on a different machine from the source.

Create a database backup plan

It is important for you to develop a database backup plan that you can follow consistently and effectively. Your database backup plan may focus solely on backing up your databases or may be integrated with backing up your enterprise systems as part of a corporate backup, replication, and recovery strategy.

To create an effective database backup plan, you need to:

- Identify database files to back up.
- Determine whether to use a full-only or full-and-incremental backup strategy.
- Determine whether to perform online or offline backups. This guide covers online backup only.
- Determine which backup tools to use.

In addition, you need to choose which type of backup media to use (disk, tape or both). As a best practice, make sure that your backup file and directory names include date and, if required, time information.

This guide discusses the procedure to back up your database online in the Set Up OpenEdge Replication databases section.

2

Plan for OpenEdge Replication

In this section, you learn how to plan for OpenEdge Replication. First, you learn what you must do to prepare your production environment for replication. Then, you learn about the decisions to make about how replication should work, so that you can configure OpenEdge Replication accordingly.

Note: Most of what you will learn in this section applies to both Replication Sets and single-target replication.

For details, see the following topics:

- Prepare your production environment for replication
- Decide how you want replication to work

Prepare your production environment for replication

After you identify a database that you want to replicate, you must consider some requirements to prepare your production environment for OpenEdge Replication:

- Is there sufficient AI space?
- Is there a live communication channel between the Source database and the Target databases?
- Are there sufficient system resources?
- Is there sufficient network bandwidth?

Understand OpenEdge Replication after-image requirements

To use OpenEdge Replication, you must activate after-imaging on the source database. Additionally, if you define a Replication Set, you must activate after-imaging on your target databases as well. When you enable after-imaging, the database engine writes notes containing a description of all database changes to the after-image logs. You can use this information generated by after-imaging to recover a database that was damaged when a failure caused the loss of either the database itself or its primary recovery (before-image) area.

Before you start using OpenEdge Replication, it is recommended that you calculate how much after-image volume, on average, your database currently generates. You can then use this average to estimate the network bandwidth that establishing replication for the database will require. After-image data plays the largest part in determining how much bandwidth you need.

Although you cannot calculate exact data volume, since it is dependent on the type of transactions within the application and how much activity is going on against the database, you can arrive at an estimate if you know the average amount of after-image volume on your system.

For more general information about database after-imaging, see *Manage the OpenEdge Database*.

Calculate fixed after-image extent size

To estimate how much AI space is required, you must first calculate the database's current AI volume. There are three methods to determine how much AI data your database is generating—you can use AI virtual system tables (VSTs), the RFUTIL command, or the PROMON command. In this guide, you learn how to use the PROMON command. To learn about the other two methods, refer to the *Progress Use Database Replication*.

In practice, you would not want to collect just one sample. You would want multiple samples over several days and then calculate the average to get a more accurate estimate of AI volume.

Note: None of the methods to calculate AI data volume considers spikes in AI activity.

Note: If your production database is not AI-enabled, then you can use before-image (BI) VSTs to gather the required data. However, to implement replication, you must enable AI on your production database.

Assume that the before-image block size is the same as the after-image block size. Before-image writes can be used to determine approximately how many after-image blocks will be written. To understand the network bandwidth requirements for each Replication Environment during peaks of high activity, the maximum network bandwidth can be estimated with the following formula:

```
Replication_Bandwidth = 1.5 * BI (writes per second at peak time) * (BI block
size) = number of bytes / per second
```

To find this information use the following procedure:

1. **PROMON** > **R&D** > **2 (Activity Displays)** > **5 (BI Log):**

```
Activity Displays Menu
1. Summary
2. Servers
3. Buffer Cache
4. Page Writers
5. BI Log
```

2. Gather metrics on the Total BI writes per second:

Activity: BI Log			
Total	Per Min	Per Sec	Per Tx
Total BI writes 151	906	**150.10**	0.70

3. Calculate the Replication Bandwidth:

   Assuming the BI block size is 16 KBytes

   `Replication_Bandwidth = 1.5 * 150.10 writes/sec * 16 KBytes/write = 3,602.4 KBytes/sec`

   OpenEdge Replication sends approximately 3.5 MB/sec of data to the target database during peak activity.

4. To calculate the communications bandwidth requirement you need to convert to bits/second, for example:

   `3,602.4 KBytes/sec * 8 bits/Byte = 28,819.2 Kbits/sec = 28.14 Mbits/sec or 28.14 Mbps`

Determine OpenEdge Replication network bandwidth

There is additional replication overhead that is required to transfer the after-image blocks over the network to the target system. This additional overhead is estimated at ten percent over the amount of the message volume sent between the server and agent and consists of header and control information for replication.

Items in this calculation include, but are not limited to:

- Packaging of blocks. Replication adds overhead to each AI block.
- The database default after-image block size of 8K. Changing the after-image block size to 16K means fewer messages are sent, which lowers the network activity by approximately five percent.
- Standard communication between the OpenEdge Replication agent and the OpenEdge Replication server.

Determining network bandwidth can be difficult. Consider that a T1 line provides approximately 1.544 megabits per second of throughput. This value represents the theoretical limit; however, the value is typically somewhat less than this due to routers, hubs, and switches.

The presence of after-image data is the most significant determinant in how much bandwidth is needed. Exact data volume cannot be accurately calculated, as it depends on the type of transactions within the application and how much activity is going on against the database. However, it can be estimated if you know the average amount of after-image volume on your system. Use the following formula to help you calculate the values for capacity planning:

```
AID x AIRM = AIDR
AIDR x AIRO = AIDRO
AIDRO X (PRD) = AIDROP
```

AID	The total size of all the after-image blocks.
AIRM	The value 1.5; the multiplier for additional after-image notes due to OpenEdge Replication.
AIDR	The number of after-image blocks with OpenEdge Replication.
AIRO	The value 1.1; the multiplier for OpenEdge Replication overhead.
AIDROP	The number of after-image blocks with OpenEdge Replication and overhead.
(PRD)	The period of time, as follows: • 1, if samples are 60 minutes and you want per-hour values • 1/60, if samples are 60 minutes and you want per-minute values • 24, if samples are 1 hour and you want per-day values • 60, if samples are 1 minute and you want per-hour values • 1/24, if samples are 24 hours and you want per-hour values • 1/1440 (which is 24 * 60), if samples are 24 hours and you want per-minute values

Example

- Consider the following example, where:
  - AID is 89MB.
  - Sample length is 60 minutes.
  - PRD is 1.

```
89MB x 1.5 = 133.5MB
133.5MB x 1.1 = 146.85MB
146.85MB x 1 = 146.85MB
```

  In this case, you would send approximately 146.85 megabytes of data to the target database per hour.

- In the example that follows:
  - AID is 360 MB.
  - Sample length is 24 hours.

- PRD is /1440.

```
360MB x 1.5 = 540MB (566, 231, 040 bytes)
5540MB X 1.1 = 594MB (622, 854, 144 bytes)
594MB/1440 = 422 KB (432, 537)
```

In this example you would send approximately 422 kilobytes of data to the target database per minute.

Decide how you want replication to work

The processes of replication your database with OpenEdge is dynamic in nature. There are many ways in which you can choose to replicate your database. This is by design to accommodate as many enterprises as possible. However, this quick start guide focuses on a specific predetermined path to showcase an example of replication and database high availability.

Choosing the replication agent startup mode

Whether you are implementing a Replication Set or single-target replication, you can choose between two startup modes for the replication agent—deferred agent startup and immediate agent startup. This guide uses deferred agent startup.

Note: Deferred agent startup requires asynchronous replication.

Deferred agent startup mode—The deferred agent startup mode enables the Source and its replication server to start without connecting to the replication agents, provided you have implemented asynchronous replication. This mode is useful if you want to create your Targets from an online backup of the Source. This guide uses the deferred agent startup mode.

Once replication starts:

1. If the replication server cannot connect to its agents on the first attempt, the server will go into the deferred agent startup mode, allowing Source database activities.
2. While in the deferred agent startup mode, the replication server makes a connection attempt every five minutes until the agents are connected or the mode is timed out.
3. When the server is connected to the agents, initialization and synchronization take place. Source database activity is paused until synchronization is complete.

Immediate agent startup mode—The immediate agent startup mode, on the other hand, does not allow any Source or Target activity until the replication server and agents complete their startup and initialization activities. To configure this mode, you must configure synchronous replication.

Because deferred agent startup allows for online backups and minimizes downtime, it is the recommended mode for starting your replication agents.

Choose an online or an offline backup of the source database

As you plan for OpenEdge Replication, you identify the database you will use as a source. You then use a backup of that database to create the target database.

You can choose to use either an online backup or an offline backup of the source to create the target database. The steps you follow to set up the source database and enable it for replication are, however, dependent upon the backup method that you choose.

This guide uses an online database backup because it reduces downtime and brings the database closer to high availability. Online database back up is discussed in more detail later in this guide.

Asynchronous replication method

Data can be replicated from the Source to the Targets asynchronously. In asynchronous replication the replication server keeps sending AI blocks to the replication agent as updates are made to the Source. Thus, there is no wait time between the end of one transaction and the beginning of another.

Versions of OpenEdge prior to 12.0 could use synchronous or asynchronous replication. However, synchronous replication is much slower than asynchronous replication. For this reason, asynchronous replication is the recommended replication method for OpenEdge versions prior to 12.0.

Choose transition properties

Finally, you need to make decisions about the following aspects of transition:

- In a Replication Set, which Target is primary and which is secondary?
- Which type of AI functionality should be implemented in the Targets?
- Should the Targets restart automatically after transition?
- What should be the Target behavior in the case of the loss of a Source?
- Should the transition be initiated manually or automatically?

The first three decisions apply to both planned downtime and the loss of a Source. The last two decisions only apply to the loss of a Source.

When you set the properties in the properties file, you configure how you want the entire replication framework to perform before, during, and after transition.

The following is an example of transition properties.

```
replication-set=1
database-role=reverse
transition-to-agents=agent1, agent2
auto-begin-ai=1
auto-add-ai-areas=0
ai-structure-file=addAI.st
recovery-backup-arguments=backup-db0-recovery.bak
restart-after-transition=1
source-startup-arguments=-pf sourcestart-args.pf -S 20000
target-startup-arguments=-pf targetstart-args.pf -S 20000
```

The following table provides a summary of the transition properties. The table lists each property in the `repl.properties` file, identifies the property type, and provides a property description.

Table 1: Transition properties

Property name	Type and length	Description
`replication-set`	integer	Defines if source and target replicas transition together. Possible values are as follows: • **`0`** — The source and targets do not transition together (the default). • **`1`** — The source and targets transition together. For more information, see 'Transition of a Replication Set' in *Use Database Replication*.
`database-role`	character[15]	The new role of the database once it is transitioned. The possible values for this property are as follows: • **`reverse`** — The role of the database is reversed: A source database becomes a target database, and a target database becomes a source database. • **`normal`** — The role of the database becomes that of a normal database; the database is no longer enabled for replication once the transition is performed. This is the default value.
`responsibility`	character[15]	This property must contain one of the following values: • **`primary`** — This database is the primary database. • **`secondary`** — This database is the secondary database. This property is currently informational.

Property name	Type and length	Description
restart-after-transition	integer	The database can be automatically restarted after transition is performed. Valid values for the property are 0 and 1. When the property is set to 1, the following *-startup-arguments properties must be supplied, or the database startup will fail: • If the database role is normal, you must specify the normal-startup-arguments. • If the database role is reversed, you must specify the source-startup-arguments and the target-startup-arguments. **Note:** The restart-after-transition property behaves differently during online transition. Setting the property value to 1 does not automatically restart the database. The value is ignored. For more information, see 'Plan for transition' in *Use Database Replication*.
start-secondary-broker	integer	Set to **1** to automatically start a secondary broker after transition.
source-startup-arguments	character[256]	If the database is transitioned to a source database, these arguments are used when the database is started. The arguments will be appended to the PROSERVE command (used to start the database). In most cases, the only argument specified here should be -pf followed by a parameter file name. For example: -pf *db-name*.source.pf
target-startup-arguments	character[256]	If the database is transitioned to a target database, these arguments are used when the database is started. The arguments will be appended to the PROSERVE command. In most cases, the only argument specified here should be -pf followed by a parameter file name. For example: -pf *db-name*.target.pf Because the database is a target, you must also specify the following arguments as an indication to the broker to listen on the TCP/IP port specified with -S *port-number* or *service-name*: -S { *port-number* | *service-name* }

Property name	Type and length	Description
`normal-startup-arguments`	character[256]	If the database is transitioned to a normal database, these arguments are used when the database is started. The arguments will be appended to the PROSERVE command. In most cases, the only argument specified here should be `-pf` followed by a parameter file name. For example: `-pf` *`db-name`*`.normal.pf`
`source-secondary-broker-startup-arguments`	character[256]	For secondary broker. See the description of `source-startup-arguments` for details.
`target-secondary-broker-startup-arguments`	character[256]	For secondary broker. See the description of `target-startup-arguments` for details.
`normal-secondary-broker-startup-arguments`	character[256]	For secondary broker. See the description of `normal-startup-arguments` for details.
`auto-begin-ai`	integer	Specify **1** to automatically begin AI after a target-to-source database transition. After-imaging can be started for a database that has AI areas.
`transition-to-agents`	character[256]	Transition to the first agent in the list when a failure occurs. If the first agent is not available, transition to the second agent in the list. If `replication-set=1`, the first agent in the `transition-to-agents` list is selected to become the new source. A valid value for this property is any configured agent name; separate the names by a comma if you are listing more than one. For example: `agent1,agent2` **Note:** If both `dr-transition-to-agents` and `transition-to-agents` are defined, `dr-transition-to-agents` is used in *recovery* transition and `transition-to-agents` is used for *failover* transition. If `dr-transition-to-agents` is not defined,`transition-to-agents` is used for both types of recovery.

Property name	Type and length	Description
dr-transition-to-agents	character[256]	**For Recovery Transition only.**Transition to the first agent in the list when a failure occurs. If the first agent is not available, transition to the second agent in the list. If replication-set=1, the first agent in the transition-to-agents list is selected to become the new source. A valid value for this property is any configured agent name; separate the names by a comma if you are listing more than one. For example: agent1,agent2 **Note:** If both dr-transition-to-agents and transition-to-agents are defined, dr-transition-to-agents is used in *recovery* transition and transition-to-agents is used for *failover* transition. If dr-transition-to-agents is not defined,transition-to-agents is used for both types of recovery.
auto-add-ai-areas	integer	This property instructs the transition process to automatically add AI areas to the database if: • It is transitioned to a source database. • There are currently no AI areas for the database. If the value specified for the property is **0**, AI areas are not added to the database. If the value specified is **1**, AI areas are automatically added to the database by using the structure file specified in the ai-structure-file property.
ai-structure-file	character[256]	Name of the structure file that contains the list of AI areas to add.

Property name	Type and length	Description
`backup-method`	character[15]	Specifies the backup method performed before AI is enabled. The following values are valid: • **`mark`** — Mark the database backed up by using the following command: `rfutil` *`db-name`* `-C mark backedup` Marking the database as backed up does not allow future AI extents to be used when recovering from a disaster. • **`full-offline`** — The database will be backed up offline by using the PROBKUP utility. The backup is performed in two steps: The first backup is a full backup, which is performed before AI is enabled for the database. The second backup is an incremental backup, which is performed after AI is enabled and after the role of the database is changed. • **`full-online`** — An online backup is performed after the database is restarted, which occurs after the database has been transitioned. **Note:** The `backup-method` property behaves differently during online transition. For more information, see 'Offline backup and online failover transition' in *Use Database Replication*.
`backup-arguments`	character[256]	The arguments required for the full online and offline backups that are performed for the database. At a minimum, the target file or device must be specified in these arguments for both online and offline backups. To avoid overwriting a backup, do not use the same target file or device for both the backup and the incremental backup. Do not use backup validation parameters (such as `-vp` and `-vf`). The arguments should begin with *`device-name`*.
`incremental-backup-arguments`	character[256]	The arguments required for the offline incremental backup performed after AI is enabled and the database's role is reversed. The arguments should begin with *`device-name`*.

Property name	Type and length	Description
`recovery-backup- arguments`	character[256]	The arguments required for the full online and offline backups that are performed for the database. At a minimum, the target file or device must be specified in these arguments for both online and offline backups. To avoid overwriting a backup, do not use the same target file or device for both the backup and the incremental backup. Do not use backup validation parameters (such as `-vp` and `-vf`). The arguments should begin with *`device-name`*.
`apply-streaming-data`	integer	**For auto-transition only.** If Replication AI Streaming is enabled, and there is data in the RSB cache at the time of transition, indicate if the data should be applied or ignored before transition. The following values are valid: • **1** — Apply all data in the RSB cache before transition. • **0** — Do not apply any data in the RSB cache before transition. **Note:** Replication AI Streaming is enabled by default.

3

Configure the property files

Configure the source database property file

Before you start OpenEdge Replication on the source database, you must configure the replication server. You configure the server by saving and then editing the sample source properties file provided when you install OpenEdge Replication as part of OpenEdge. Each of the properties is listed and described in Server properties.

To configure the source database property file:

1. Make a copy of the sample `source.repl.properties` file, which is located by default in `OpenEdge-install-dir\properties`.
2. Name this new file `db-name.repl.properties`, where db-name is the name of your source database.
3. Place this new file in the same directory as your source database.
4. To change any of the default property values, simply modify them in the property file, as follows:
   - If you are on UNIX, you can use vi, emacs, or your preferred text editor to change the properties file. You can also use OpenEdge Management or OpenEdge Explorer.
   - If you are in Windows, you can use OpenEdge Management or OpenEdge Explorer to change the properties file. You can also use Notepad.

   For information about configuring the properties file in OpenEdge Management or OpenEdge Explorer, see Configure OpenEdge Components with OpenEdge Management.

Configure the target database property file

Before you start OpenEdge Replication on the target database, you must configure the agent. You configure the agent by saving and then editing the sample target properties file provided when you install OpenEdge Replication as part of OpenEdge. Each of the properties is listed and described in Agent properties.

Use the following steps to configure the target database property file:

1. Make a copy of the `target.repl.properties` file, which is located by default in `OpenEdge-install-dir\properties`.
2. Name this new file `db-name.repl.properties`, where db-name is the name of your target database.
3. Place this new file in the same directory as the target database.
4. To change any of the default property values, edit and save the file.

   If you are on UNIX, you can use vi, emacs, your favorite text editor, or OpenEdge Explorer to change the properties file. If you are in Windows, you can use Notepad or OpenEdge Explorer to change your properties file. You can also use OpenEdge Management if you have purchased a license.

   For more information about configuring the properties file in OpenEdge Management or OpenEdge Explorer, see Configure OpenEdge Components with OpenEdge Management.

Validate the property files

The DSRUTIL properties validation utility enables you to validate the replication properties of replication-enabled source and target databases and confirm running status of target database brokers.

When run against the replication source database, the utility performs the same validation made during startup of the replication server, validating all properties listed in the `[server]`, `[control-agent]`, and `[transition]` sections of the properties file. For each control-agent listed in the `[server].control-agents` property, the utility also pings the target database on the hostname and port in the associated `[control-agent]` section of the properties file.

When run against the replication target database, the utility performs the same validation made during startup of the replication agent, validating all properties listed in the `[agent]` and `[transition]` sections of the properties file. If there is a `[server]` section defined in the target database's properties file, the length of each control-agent name in the `[server].control-agents` property is validated.

The utility writes all properties file errors to both standard out and the database log file.

The utility can be run while the database is online or in single-user mode.

Syntax

The command for the utility is as shown:

```
dsrutil db-name -C validate
```

Parameters

db-name	The name of the database to perform the action on. The name of the database must be the first argument and must be a valid name.

Notes

- The utility adds the ability for the source database to report on the running status of any target database brokers using messages 19460 and 19461.

```
Database db-name for agent agent-name is online. Broker is running on
hostname:port-num. 19460
```

```
Unable to contact database db-name on hostname:port-num for agent agent-name.
19461
```

For details, see the following topics:

- Understand DSRUTIL STATUS vs -detail (or verbose) return codes

Understand DSRUTIL STATUS vs -detail (or verbose) return codes

When performing specific OpenEdge Replication server, OpenEdge Replication agent, source database, and target database requests with the DSRUTIL utility it is important to understand the differences in return codes.

When you use the command: `dsrutil dbname -C status`, only the **status** codes are returned. These are essentially a per-fixed **group status code**. For example:

```
1101 : DSRUTIL source -C status
2101 : DSRUTIL source -C status agent1
3101 : DSRUTIL target -C status
```

When `dsrutil dbname -C status -detail` (or verbose) is used, only the **detail status** codes are returned, which are a sub-set of the group status'. For example:

```
6021 : DSRUTIL source -C status -detail
3049 : DSRUTIL source -C status agent1 -detail
1063 : DSRUTIL source -C status agent2 -detail
```

It is important to understand the difference between these two sets (a set and a sub-set) as together they provide a fuller picture of the replication total status. For example, the status group `101 Processing` in the previous example provides a finer granularity of what processing is taking place. For example:

- RPLS 6005 | RPLA 3048 Startup Synchronization
- RPLS 6021 | RPLA 3049 Normal Processing
- RPLS 6004 | RPLA 3050 Recovery Synchronization

The following list provides this detail grouped by focus, source, target, or source to agent). Using `DSRUTIL -C status verbose` provides both the **status code and description** to record status pages.

Server to Agent initial connect status codes

Status	Description
1001	Initial Connection
1002	Initializing
1003	Target in Quiet Point
1032	Initial Connection Failed
1033	Recovery Failed
1034	Invalid Target Database Configuration
1035	Agent Failed
1036	Agent is Ignored
1037	Agent is Stopped
1038	Agent is Terminated
1063	Agent is Ended

Replication Server

Status	Description
1100	Connecting status
1101	Processing status
1102	Stalled status
1103	Pre-transition status
1104	Listening status
1105	Transition status
1199	Inactive
1255	Unknown
6001	Server Initialisation
6002	Connecting to Agent(s)
6003	Configuring Agent(s)
6004	Recovery Processing
6005	Startup Synchronization
6006	Initiating Agent to Agent Connections
6021	Normal Processing
6022	Performing Transition
6060	Server is ended

Control Agent

Status	Description
2100	Connecting status
2101	Processing status
2102	Stalled status
2103	Pre-transition status
2104	Listening status
2105	Transition status
2080	Pre-transition
2081	Applying After-image Extent
2082	Transitioning
2083	Listening
2199	Inactive
2255	Unknown

Agent

Status	description
3100	Connecting status
3101	Processing status
3102	Stalled status
3103	Pre-transition status
3104	Listening status
3105	Transition status
3048	Startup Synchronization
3049	Normal Processing
3050	Recovery Synchronization
3051	Being Transitioned
3052	Target Database in Quiet Point
3053	Target Database is in a BI stall
3054	Target Database is in an AI stall
3055	Online backup of Target Database
3199	Inactive
3255	Unknown

Example

Here are some examples of status before connection:

```
$ DSRUTIL [source] -C
"STATUS:" 1100 "STATE CONNECTING"
"DETAIL:" 6002 "Connecting to Agents"
"AGENT:" 1001 "Initial connection"
```

```
$ DSRUTIL [target] -C
"STATUS:" 3104 "STATE LISTENING"
"DETAIL:" 2083 "Listening"
"AGENT:" 2083 "Listening"
```

Then when connected:

```
$ DSRUTIL [source] -C
"STATUS:" 1101 "STATE PROCESSING"
"DETAIL:" 6021 "Normal Processing"
"AGENT:" 3049 "Normal Processing"
```

```
$ DSRUTIL [target] -C
"STATUS:" 3101 "STATE PROCESSING"
"DETAIL:" 3049 "Normal Processing"
"AGENT:" 3049 "Normal Processing"
```

4

Set Up OpenEdge Replication databases

You can prepare to run OpenEdge Replication using an online backup of the source database to create the target database. To enable online backups, you must implement the deferred agent startup mode, which is described in **Use the deferred agent startup mode**. You also must enable AI on the source database.

Perform the following steps on the source database:

Step 1: Create a structure file

Once you shut down the source database, you must build a structure file (`.st`). Use the following command to build this file:

```
prostrct list source-db-name source-db-name.st
```

Later you will copy this structure file from the source to the target machine.

Step 2: Enable the database for after-imaging

To use OpenEdge Replication, you must enable after-imaging on the source database.

If after-imaging is already enabled for the database, skip this topic and go to Step 3: Enable the source database for OpenEdge Replication.

If after-imaging is not already enabled for the database, use the procedure that follows to enable it.

1. Create a new structure file—named, for example, source-db-name_ai.st—and edit it to add after-imaging.

   The following is a sample of the structure file:

   ```
   # addai.st
   a C:\OpenEdge\WRK\ f 1024
   a C:\OpenEdge\WRK\ f 2048
   ```

   Be sure to give the new structure file a different name from the one you created in Step 1: Create a structure file.

2. Apply source-db-name_ai.st to the source database.

   The syntax is shown:

   ```
   prostrct add source-db-name source-db-name_ai.st
   ```

3. Back up the database.

   The syntax is shown:

   ```
   probkup source-db-name source-db-backup-name
   ```

4. Begin after-imaging.

   The syntax is shown:

   ```
   rfutil source-db-name -C aimage begin
   ```

Step 3: Enable the source database for OpenEdge Replication

OpenEdge Replication requires that the source database be enabled for replication before you start the database. To enable the source database, use the following command:

```
proutil source-db-name -C enableSiteReplication source
```

Step 4: Perform an online backup

To perform an online backup, use the online and -REPLTargetCreation options, as shown:

```
probkup online source-db-name source-db-backup-name -REPLTargetCreation
```

Step 5: Configure the OpenEdge Replication server for deferred agent startup

Online backups require a server that is configured for deferred agent startup. To configure such a server, you must set the `defer-agent-startup` property to a valid time-out value (in minutes) in the server property file. In the following example, the time-out (shown in bold) is set for 240 minutes, or four hours:

```
[server]
     control-agents=agent-name
     database=source-db-name
     transition=manual
     transition-timeout=1200
     defer-agent-startup=240
```

Create the OpenEdge Replication target database

To set up OpenEdge Replication, you must restore the backup you created of the source database to use as your target database.

The information in the following sections is provided based on the expectation that you are familiar with using OpenEdge databases and their supported commands. If you are not familiar, refer to *Manage the OpenEdge Database*.

Use the following steps to create the OpenEdge Replication target database.

1. Move or copy the source backup file and the incremental backup file from the source database machine to the target database machine.
2. Copy the structure file (source-db-name`.st`) from the source database to the target machine.

   This structure file lists the physical structure of the source database. If the physical structure of your target database is different (for example, there are different drives, slices, or directories) you must edit the structure file for the target database to accurately describe the physical structure of the target database.
3. If the source database is encrypted, copy the source database key store (source-db-name`.ks`) to the target database directory and rename it with the name of the target database (target-db-name`.ks`).

   The source database keystore was created when you enabled transparent data encryption on the source database.
4. Restore the backup copy of the source database to the target machine.

   The syntax follows:

   ```
   prorest target-db-name source-db-backup-name
   ```
5. Enable the target database for OpenEdge Replication.

   The syntax follows:

   ```
   proutil target-db-name -C enableSiteReplication target
   ```

For details, see the following topics:

- Configure the database service communication area size (-pica)

Configure the database service communication area size (-pica)

The database service queue holds a list of the IDs of all the AI blocks that must be replicated to the Target database. If the queue is full, all incoming ID entries are discarded, so information about AI blocks that are yet to be replicated is lost.

To prevent this from happening, you should set the database service communication area size (`-pica`) to its maximum size (`1000000`). The system cost is very low and the penalty for having too small a value can be high.

We recommend that when you set up the Source and the Targets for replication, make sure that you include `-pica 1000000` in your list of startup parameters.

5

Start the replication

Once you set up the source and target databases and their corresponding property files, you can start OpenEdge Replication. Simply start your source and target databases with an OpenEdge Replication qualifier.

The order in which you start the databases does not matter. However:

- It is recommended that if you did not set the `defer-agent-startup` property to a non-zero value in your server properties file, you start the target database first so that the OpenEdge Replication server does not time out.
- If you do start the source database first, start the target database before the `connect-timeout` property in the `[control-agent.name]` section of the server properties file expires.

Alternatively, you can start OpenEdge Replication with the database utility dbman or OpenEdge Explorer (or OpenEdge Management, if you have a license).

Start the source database

Use the following command to start the source database:

```
proserve source-db-name -S {port| service name}
```

Note: You can start the source database with any valid PROSERVE argument. When using arguments that affect shared memory (number of users, clients per server, maximum number of servers, etc.), you should be careful to use the same arguments and values when you start the target database.

If there is a configuration error, the OpenEdge Replication server will be brought down even if there is no critical agent; and messages will be sent to the database log on both the source side and the target side. (When a critical agent exists for a server, transition starts automatically for that agent's target database during failure processing.)

The following are examples of a configuration error:

- Missing AI extents
- Improperly created target
- Large file mismatch

Start the target database

Use the following command to start the target database:

```
proserve target-db-name -S {port |service name }
```

The database broker will monitor the port or service name specified by the `-S` argument; therefore, inclusion of the argument is required. The port or service name must be the same as the port or service name parameter you specify in the OpenEdge Replication server's `repl.properties` file.

The port or service name you specify with the `-S` argument can also be used by ABL clients and the Replication server. If the broker started is configured for both ABL and SQL connections, the SQL clients can use the port and service name as well.

Only limited database connections are allowed until OpenEdge Replication completes startup synchronization.

Note: You can start the target database with any valid PROSERVE argument. When using arguments that affect shared memory (number of users, clients per server, maximum number of servers, etc.), you should be careful to use the same arguments and values you used when you started the source database.
