Progress®



Back Up and Restore a Database

OpenEdge®

Copyright

© 2021 Progress Software Corporation and/or its subsidiaries or affiliates. All rights reserved.

These materials and all Progress® software products are copyrighted and all rights are reserved by Progress Software Corporation. The information in these materials is subject to change without notice, and Progress Software Corporation assumes no responsibility for any errors that may appear therein. The references in these materials to specific platforms supported are subject to change.

DataDirect Cloud, DataDirect Connect, DataDirect Connect64, DataDirect XML Converters, DataDirect XQuery, DataRPM, Defrag This, Deliver More Than Expected, DevReach (and design), Icenium, Inspec, Ipswitch, iMacros, Kendo UI, Kinvey, MessageWay, MOVEit, NativeChat, NativeScript, OpenEdge, Powered by Chef, Powered by Progress, Progress, Progress Software Developers Network, SequeLink, Sitefinity (and Design), Sitefinity, Sitefinity (and design), SpeedScript, Stylus Studio, Stylized Design (Arrow/3D Box logo), Styleized Design (C Chef logo), Stylized Design of Samurai, TeamPulse, Telerik, Telerik (and design), Test Studio, WebSpeed, WhatsConfigured, WhatsConnected, WhatsUp, and WS_FTP are registered trademarks of Progress Software Corporation or one of its affiliates or subsidiaries in the U.S. and/or other countries.

Analytics360, AppServer, BusinessEdge, Chef Automate, Chef Compliance, Chef Desktop, Chef Habitat, Chef WorkStation, Corticon.js, Corticon Rules, Data Access, DataDirect Autonomous REST Connector, DataDirect Spy, DevCraft, Fiddler, Fiddler Everywhere, FiddlerCap, FiddlerCore, FiddlerScript, Hybrid Data Pipeline, iMail, JustAssembly, JustDecompile, JustMock, KendoReact, NativeScript Sidekick, OpenAccess, PASOE, Pro2, ProDataSet, Progress Results, Progress Software, ProVision, PSE Pro, Push Jobs, SafeSpaceVR, Sitefinity Cloud, Sitefinity CMS, Sitefinity Digital Experience Cloud, Sitefinity Feather, Sitefinity Insight, Sitefinity Thunder, SmartBrowser, SmartComponent, SmartDataBrowser, SmartDataObjects, SmartDataView, SmartDialog, SmartFolder, SmartFrame, SmartObjects, SmartPanel, SmartQuery, SmartViewer, SmartWindow, Supermarket, SupportLink, Unite UX, and WebClient are trademarks or service marks of Progress Software Corporation and/or its subsidiaries or affiliates in the U.S. and other countries. Java is a registered trademark of Oracle and/or its affiliates. Any other marks contained herein may be trademarks of their respective owners.

July 2021

Product version: Progress OpenEdge 12.4

Updated: 2021/06/28

Table of Contents

Preface....7

Creating a database backup plan....9

Identifying database files to back up....10
Determining whether to use a full-only or full-and-incremental backup strategy....12
Guidelines for determining whether to use a full-only or full-and-incremental backup strategy..13
Archiving database backups....14
Determining whether to perform online or offline backups....15
Online backups....15
Offline backups....16
Determining which backup tools to use....16
OpenEdge RDBMS backup tools....17
OS or third-party backup tools....18

Backing up a database using the PROBKUP utility....21

Performing full backups....23
Performing incremental backups....23
Backing up optional database files....24
Truncating the database log file using PROLOG....25
Procedure: Monitoring database backup status....25

Backing up a database using OpenEdge Management....27

Procedure: Creating a database backup job....30
Procedure: Running a database backup job....31

Backing up a database using alternative methods....33

Backing up a database online using OS disk mirroring....34
Procedure: Backing up a database online using OS disk mirroring....34
Backing up a database offline using OS or third-party backup tools....36
Procedure: Backing up a database offline using OS or third-party backup tools....37

Verifying a database backup....39

Using PROREST to verify a database backup....40
Procedure: Using OpenEdge Management to verify a database backup....40

Restoring a database....43

Rules for restoring a database....45
Using PROREST to restore a database....45
Generating a storage area description....46
Procedure: Using OpenEdge Management to restore a database....46

Preface

Purpose

This guide explains how to back up and restore OpenEdge® databases. The procedures in this guide help you choose the type of backup, decide which backup tools to use, and monitor backup progress. This guide assumes that you are familiar with the OpenEdge RDBMS planning concepts discussed in *OpenEdge Getting Started: Database Essentials*.

Audience

This book is intended for OpenEdge database administrators and application developers.

For the latest documentation updates go to https://docs.progress.com.

Organization

Creating a database backup plan on page 9

Explains how to identify files to back up. Helps you determine backup strategy, tools, and whether to perform online or offline backups.

Backing up a database using the PROBKUP utility on page 21

Explains how to use PROBKUP to perform full or incremental backups online or offline.

Backing up a database using OpenEdge Management on page 27

Teaches you how to create and run database backup jobs.

Backing up a database using alternative methods on page 33

Explains how to back up a database online using OS disk mirroring or offline using OS or third-party backup tools.

Verifying a database backup on page 39

Explains how to verify a database backup using PROREST or OpenEdge Management.

Restoring a database on page 43

Lists rules for restoring a database, then explains how to restore using PROREST or OpenEdge Management.

Documentation conventions

See Documentation Conventions for an explanation of the terminology, format, and typographical conventions used throughout the OpenEdge content library.

1

Creating 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 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.
- Determine which backup tools to use.

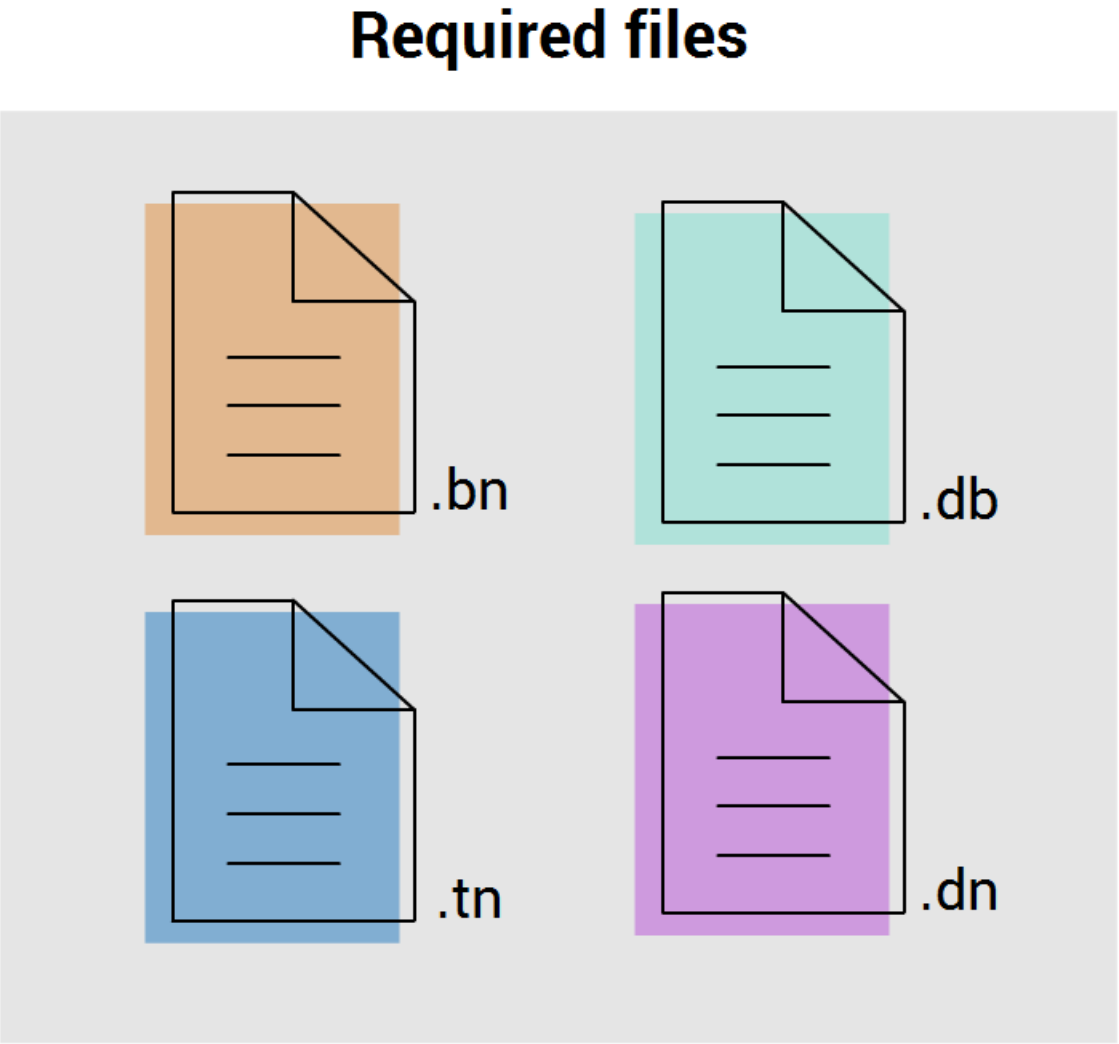
In addition, you need to choose which type of backup media to use (disk, tape or both). As a best practice, make sure that your backup file and directory names include date and, if required, time information.

Next, you will learn about each of these tasks.

For details, see the following topics:

- Identifying database files to back up
- Determining whether to use a full-only or full-and-incremental backup strategy
- Determining whether to perform online or offline backups
- Determining which backup tools to use

Identifying database files to back up



For each database, there are two groups of database files that you must back up. They are:

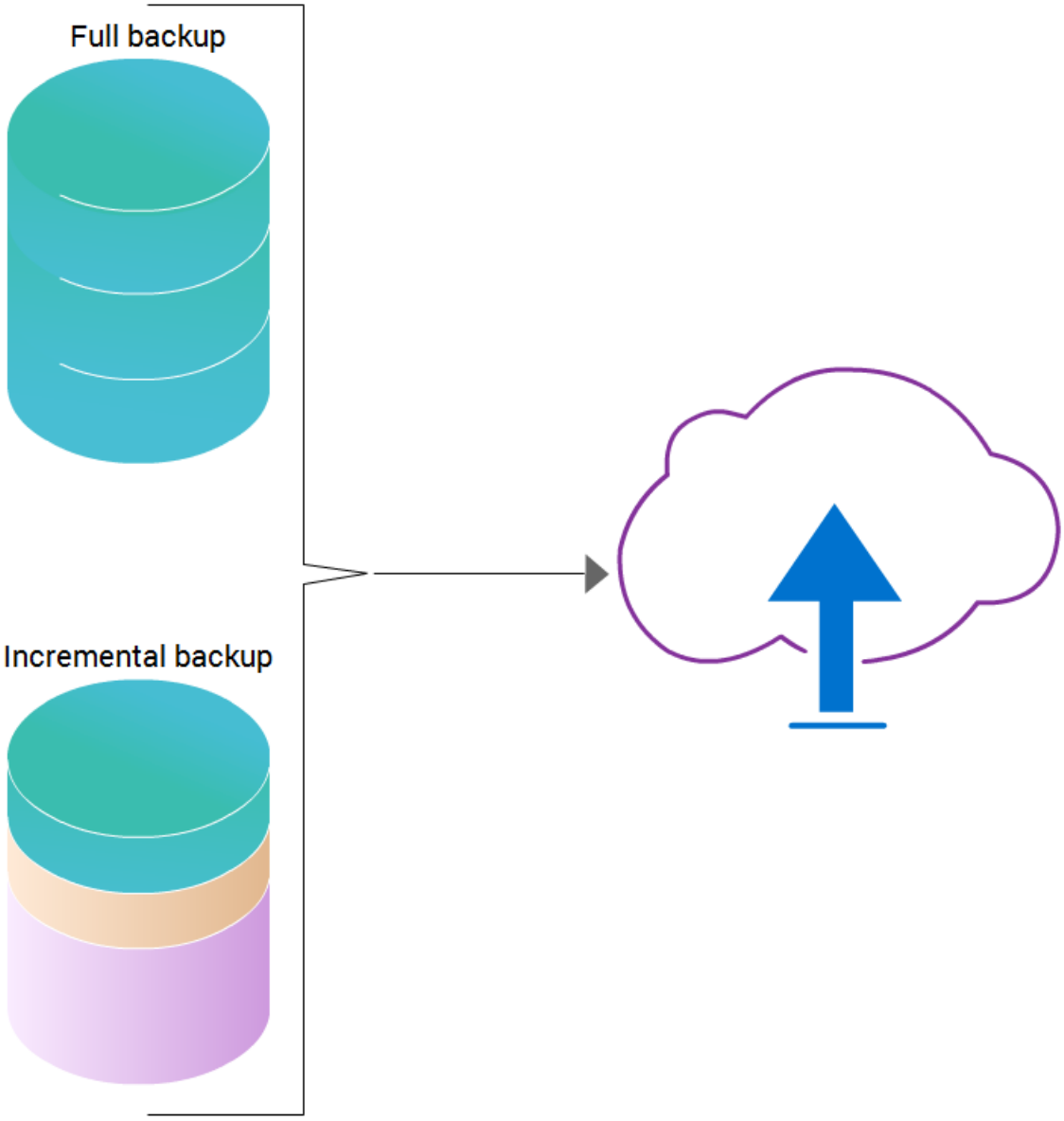
- Required files—You must back up these files as a unit because they are required to recover a consistent database. The PROBKUP utility and OpenEdge Management back up only these files. Required files include:
  - Database structure (.db) file
  - Before-image (.bn) files
  - Data (.dn) files
  - Transaction log (.tn) files (if you are using two-phase commit)
- Optional files—You must back up these files using an OS or a third-party backup tool. Optional files include:
  - Structure description (.st) file

- Database log (.lg) file
- After-image (.an) files (if you are using after-imaging)
- Key store (.ks) file (if you are using Transparent Data Encryption)

In addition to the database files, you should also back up any other files that, together with your database, make your applications work. Such additional files include:

- OpenEdge ABL code files
- Parameter files
- Procedure library files
- Properties files (conmgr.properties, ubroker.properties)
- Networking files
- OS configuration files

Determining whether to use a full-only or full-and-incremental backup strategy



A key decision you need to make for your database backup plan is whether to perform full-only backups or full-and-incremental backups.

The following table lists the two types of backup with their description:

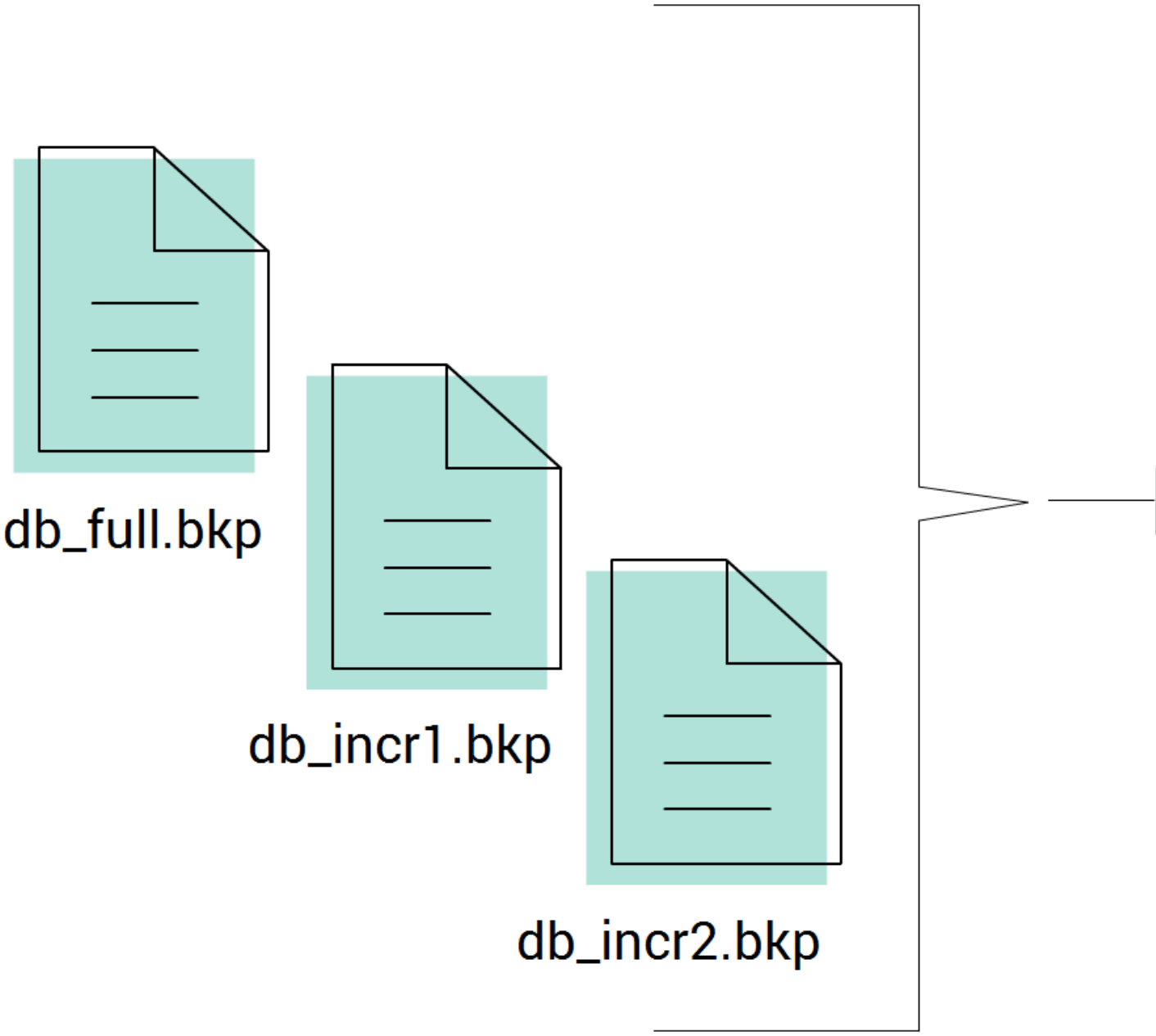
Type of backup	Description
Full backup	All database blocks in the required database files that contain application data are backed up, regardless of when they were modified. The first time you back up your database, it must be a full backup. Later, you can perform incremental backups.
Incremental backup	Only those database blocks in the required database files that have been modified since the last full or incremental backup are backed up.

Guidelines for determining whether to use a full-only or full-and-incremental backup strategy

To determine whether to use a full-only or full-and-incremental backup strategy, follow these guidelines:

- There are trade-offs between full-only and full-and-incremental backups:
  - If you choose to perform only full backups, restoring the database will be faster. However, each full backup will take longer to perform compared to an incremental backup.
  - If you choose to perform full-and-incremental backups, restoring the database will take longer because first you must restore the last full backup and then restore all subsequent incremental backups. However, each incremental backup will take less time to perform and have less impact on performance.
- Given the low cost of storage space and the ease and convenience of restoring a full backup, consider performing full backups of your database if possible.
- If you choose to perform only full backups, then do it at least daily.
- If you choose a full-and-incremental backup strategy, then you should still perform a full backup on a regular basis so that there are not too many incremental backups between full backups.
- Frequent backups give you multiple points of recoverability from which you can recover a database if required. If you discover corruption, you can restore the database up to the point that predates the corruption.
- If you cannot afford to lose data, then you should enable after-imaging.

Archiving database backups



Whether you choose to use a full-only or a full-and-incremental backup strategy, you should maintain an archive of backups. Archiving database backups gives you an additional layer of defense in case your backups are corrupted or lost.

Follow these guidelines to archive backups:

Keep at least 10 full backups. For example, keep daily backups for two weeks, weekly backups for at least two months, and monthly backups for a year.

Keep the archive of backups in a separate location from the database and from the most recent backups. Then you can still recover data if you lose the database or the most recent backups.

Determining whether to perform online or offline backups

Another key decision that you must make is whether to perform online or offline backups.

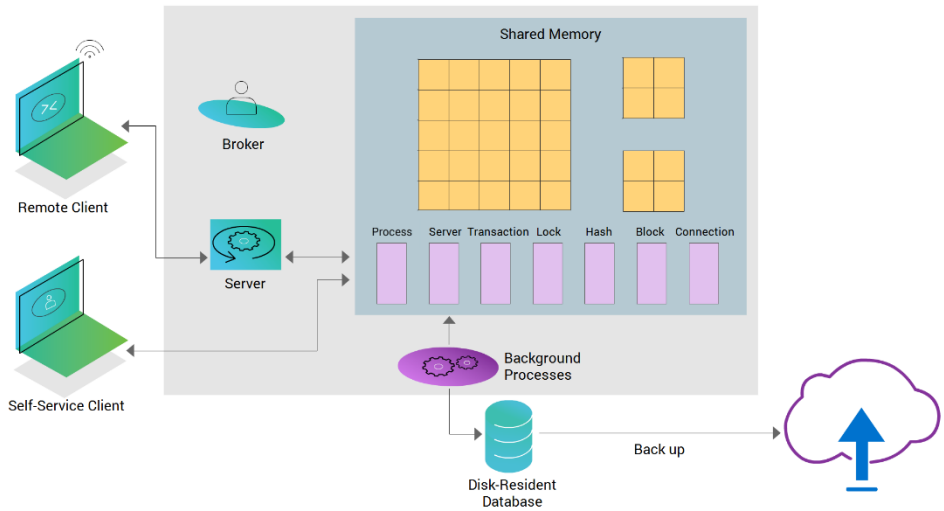
The only difference between an online and an offline backup is uptime:

If …	Then …
Your database needs to be up and running all the time,	Back up the database online.
Your database can be taken down and the offline backup time fits within the downtime window,	You can back up the database offline.

You can perform full and incremental backups either online or offline.

Next, you will learn more about online and offline backups.

Online backups



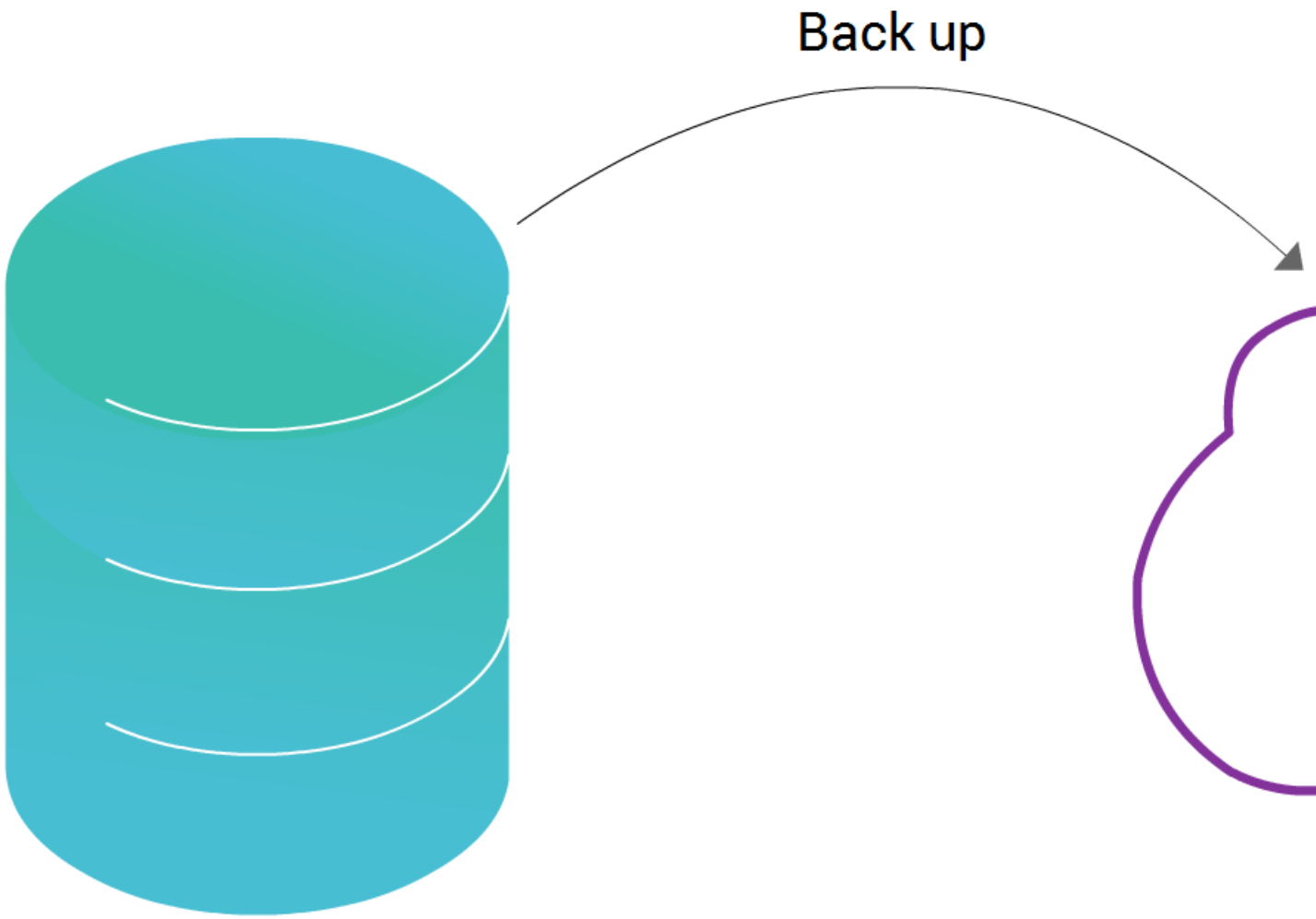
To support a 24x7 business environment, you can perform online backups while users are still accessing a database. Your applications can continue to use the database while you are backing it up online.

When you perform an online backup, it is important to take into account the following:

- You can perform an online backup only when the database is running in a multi-user mode—you cannot perform online backup in a single-user mode.
- During an online backup, all database activities are paused while the BI files are being backed up. To minimize the duration of the pause, you should ensure that your backup media is properly prepared before starting an online backup.

- If the database is enabled for after-imaging, Progress® OpenEdge® RDBMS writes to the current AI extent until it becomes full, and then switches over and writes to the next empty AI extent. So, ensure that empty AI extents are available before starting an online backup. However, if you use automatic AI file management, you do not have to do this; the automatic AI file management utility does it for you.

Offline backups



If your business environment allows you to have regular downtime for database maintenance tasks and the offline backup time fits within the downtime window, you can choose to perform offline backups.

To do so, you must disconnect all users from the database, and then shut it down. Then, you can back up the database files.

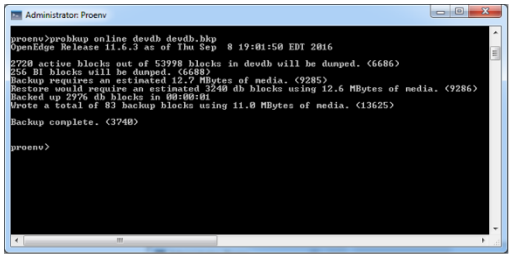
Determining which backup tools to use

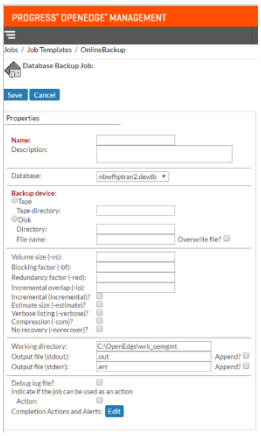
As part of your database backup plan, you need to determine which backup tools to use. You have two choices:

- OpenEdge RDBMS backup tools

- OS or third-party backup tools

OpenEdge RDBMS backup tools





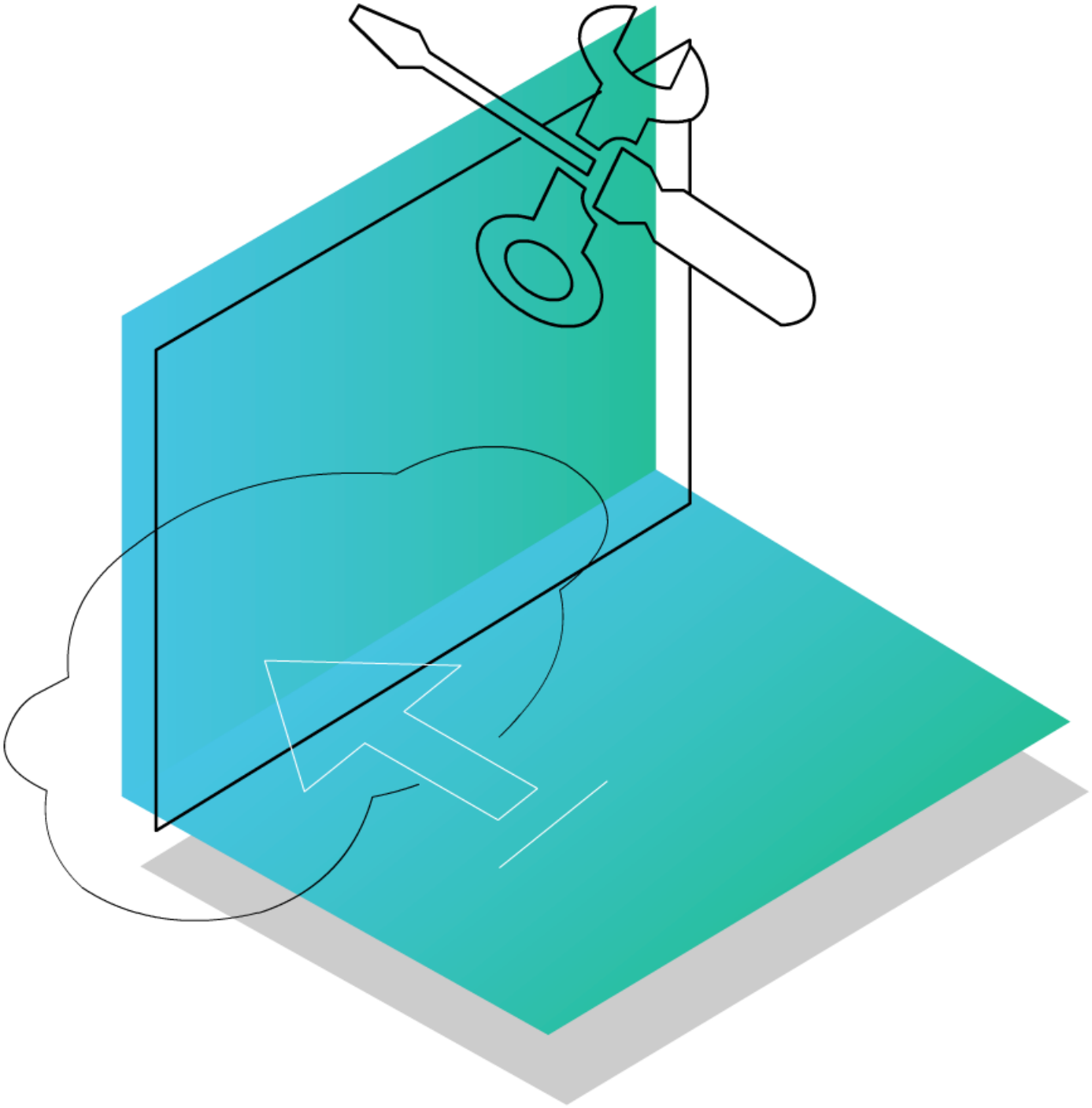
The PROBKUP utility and OpenEdge Management enable you to back up all the required files of a database.

The benefits and drawbacks of using PROBKUP or OpenEdge Management are:

Benefits	Drawbacks
• You can perform online and offline, as well as full and incremental backups. • Users can access the database during an online backup. • They back up all required files using the .db file. • Before backing up the files, they check that each database block is in the required format. • They back up only the used database blocks; they do not back up allocated but unused (empty) blocks. • They only compress but do not back up the free blocks. Free blocks are empty and can become a sequence, record, or index block. They reside below the high water mark. below the high water mark. The high water mark indicates the maximum amount of space that is currently used within a storage area. of a storage area. • They mark the database as backed up. • The backup file can be verified and restored to a new database or to a new location, using the PROREST utility.	• You can back up required files only. You must back up optional files using your OS or third-party backup tools. • The backup runs slower than hardware-based backups. • All database activities are paused while the BI files are being backed up online.

Note: After you complete backing up the required files of a database, you need to back up the optional files of the database, using OS or third-party backup tools. It is recommended that you back up the optional files every time you perform a database backup, whether full or incremental, online or offline.

OS or third-party backup tools



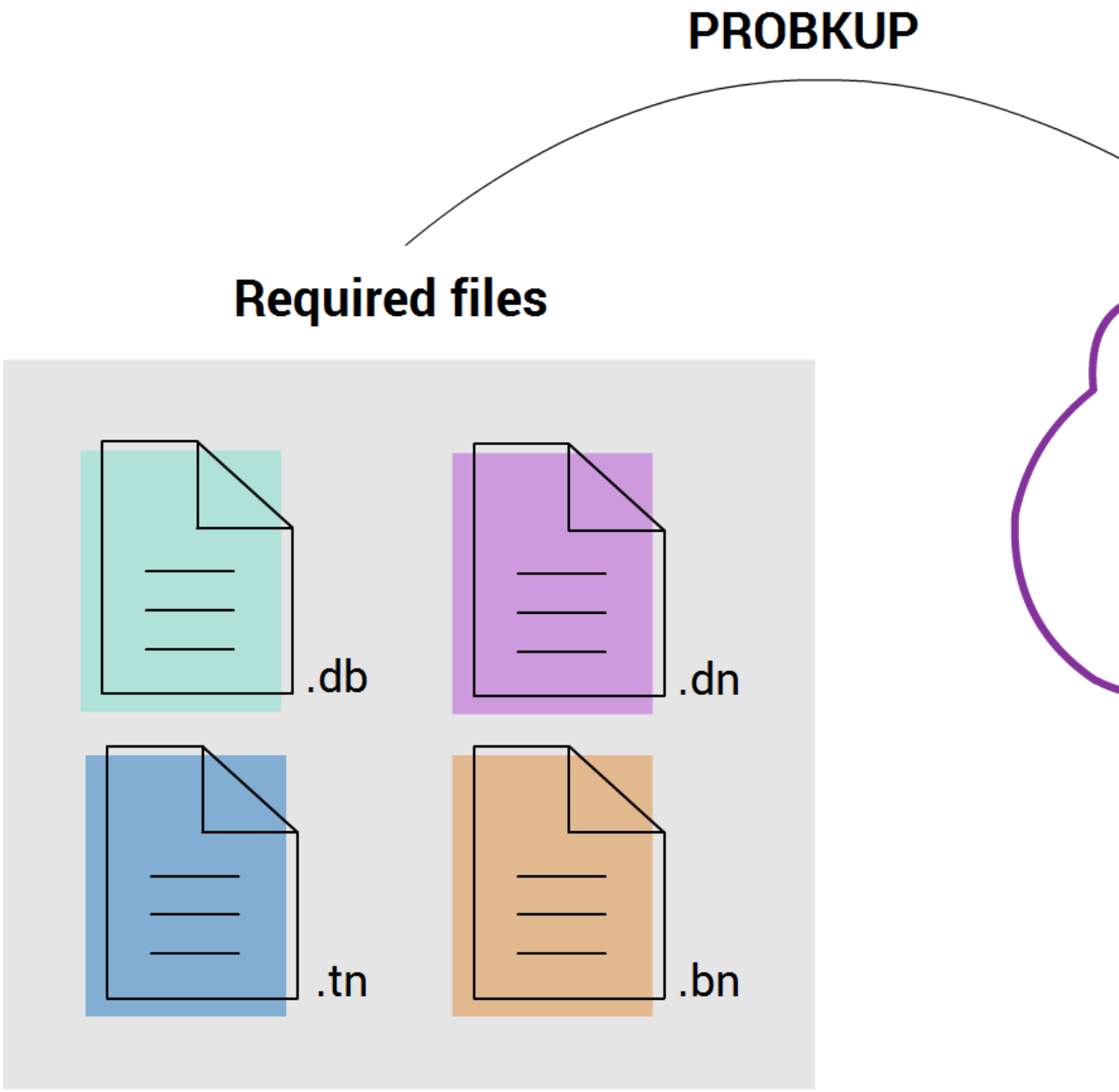
If you use OS mirroring, or OS or third-party backup tools for your corporate backup and recovery strategy, you can use the same tools to back up your databases.

The benefit and drawbacks of using these tools are:

Benefit	Drawbacks
• You can back up all the files of a database.	• These tools can be cumbersome to use. If you move, add, or delete extents, you have to keep a track of their locations and back them up manually. • You cannot do incremental backups. You must back up all the files or none at all. • These tools do not check the integrity of each database block. • These tools copy everything in a file, which requires more time and backup media.

2

Backing up a database using the PROBKUP utility



The PROBKUP utility enables you to back up all the required files of a database. Using PROBKUP, you can perform:

- Online full backups
- Offline full backups
- Online incremental backups
- Offline incremental backups

For details, see the following topics:

- Performing full backups
- Performing incremental backups
- Backing up optional database files
- Procedure: Monitoring database backup status

Performing full backups

You can use the PROBKUP utility to perform full backups of a database.

Where:

To perform an offline backup, you omit the online keyword.

Here is an example of how to perform an online full backup of the sports database and store the backup file in the backups directory:

```
probkup online sports ..\backups\sports.bkp
```

Performing incremental backups

You can also use the PROBKUP utility to perform incremental backups. If you use tapes for storing database backups, you should consider using the incremental overlap factor parameter in your PROBKUP command. The overlap factor determines the redundancy among incremental backups in a backup series of a database. Before you can perform incremental backups, you must ensure that you have already performed a full backup of a database.

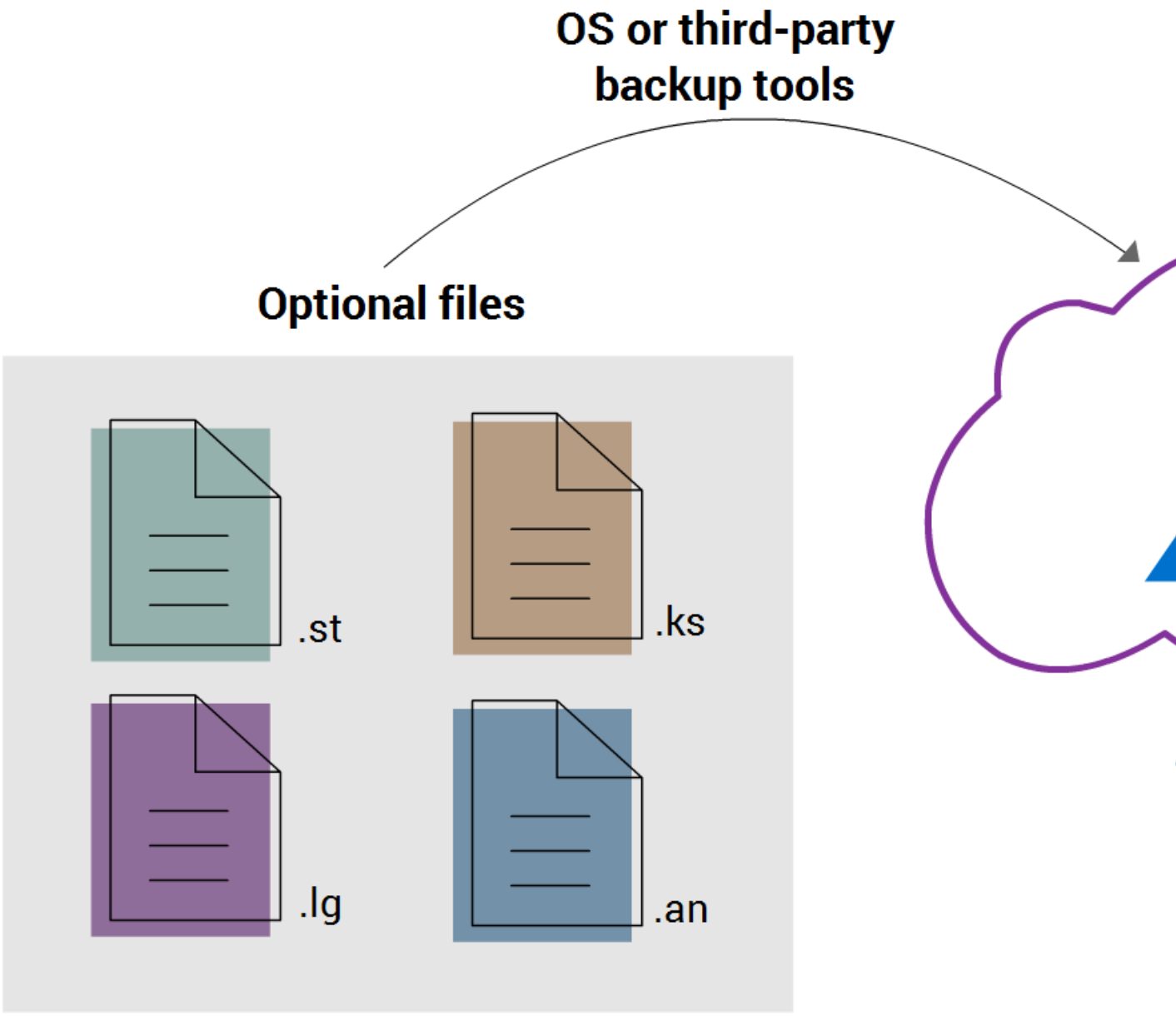
The command to perform an incremental backup of a database is:

```
probkup [online] db-name incremental backup-filename
```

Assume you have already performed an online full backup of the sports database. Here is an example of how to take an online incremental backup of sports and store the backup file in the backups directory:

```
probkup online sports incremental ..\backups\sports_incr1.bkp
```

Backing up optional database files



Recall that after you back up the required files of a database using PROBKUP, you should back up the optional files, using OS or third-party backup tools.

Once you back up the optional files of a database, you should also:

- Truncate the database log (.lg) file using the PROLOG utility.
- Back up the AI files to another location, if after-imaging is used.

Truncating the database log file using PROLOG

OpenEdge RDBMS uses the database log (.lg) file to hold startup parameter settings and to log significant database events such as startups, shutdowns, and system error messages.

As your database is being used, the log file expands, and can become quite large. So after backing up the optional files, you should use the PROLOG utility to truncate the log file.

The command to truncate the database log file online or offline is:

```
prolog db-name [–online]
```

Assume you have taken an online full backup of the sports database. Here is an example of how to truncate the sports.lg file online:

```
prolog sports –online
```

Procedure: Monitoring database backup status

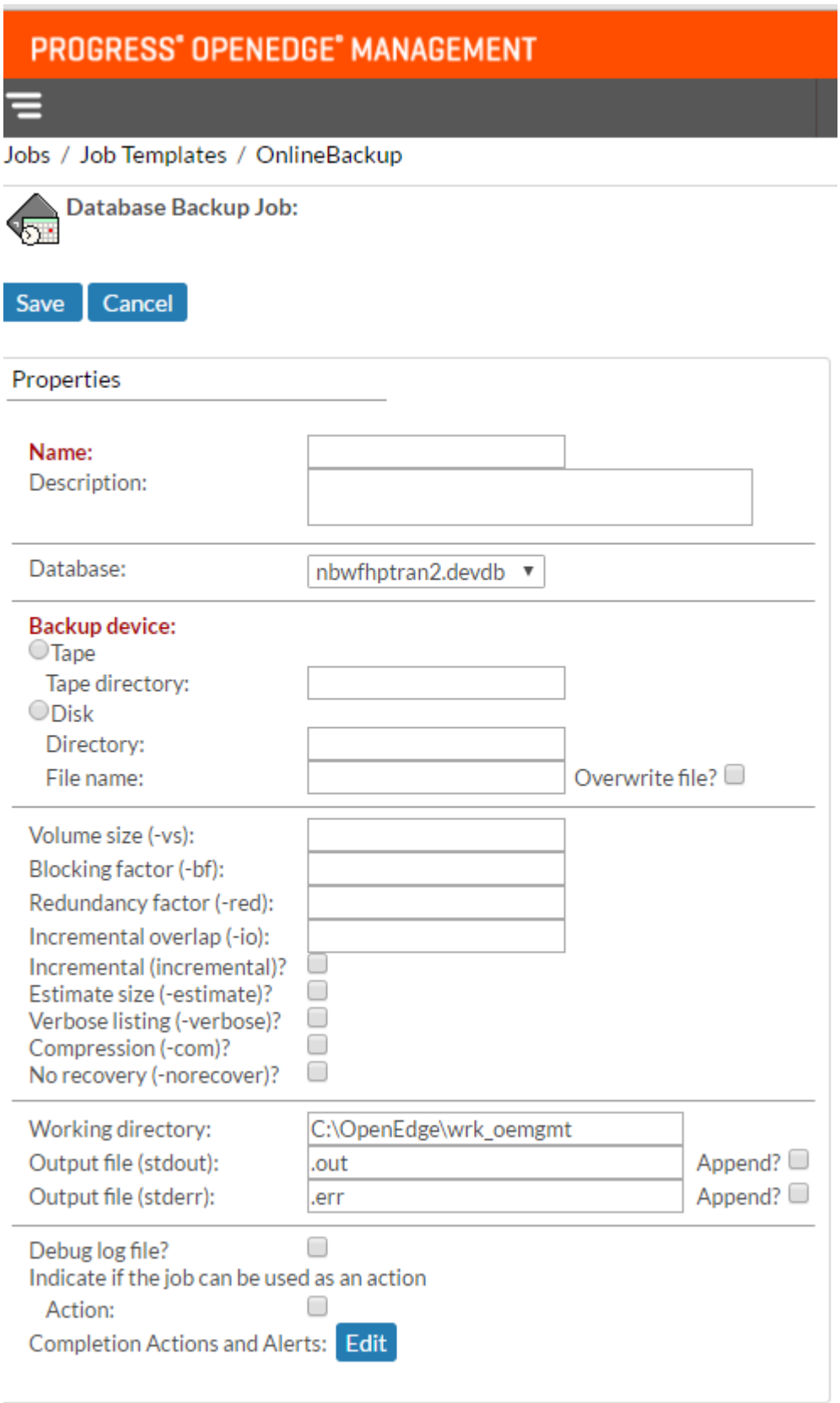
If you implement a comprehensive backup and recovery strategy, you will know when the last time a full or incremental backup was performed on a database. However, if you are brought in to manage a database that has been up and running for some time, you can use PROMON to monitor the status of the most recent full and incremental backups for a database. With this information, you can decide whether to perform full or incremental backups.

Follow these steps to monitor the status of a running database using PROMON:

Step	Action
1	Invoke PROMON on your running database: `promon db-name`
2	At the “Enter your selection:” prompt, enter 7 (for Database Status). Information about the database status such as the most recent full and incremental backups is displayed.
3	Enter q to return to the main PROMON prompt.
4	At the “Enter your selection:” prompt, enter q to exit PROMON.

3

Backing up a database using OpenEdge Management



If you have OpenEdge Management, you can use it to back up the required files of a database.

Some advantages of using OpenEdge Management are:

- You can backup databases on local and remote machines.
- You do not have to create and maintain database backup scripts, whether your databases reside in UNIX or Windows.
- You can schedule the database backup process to start automatically.

To back up a database, you create and run a backup job. OpenEdge Management provides job templates to take offline and online backups.

Next, you will learn how to use a job template to create and run a backup job.

For details, see the following topics:

- Procedure: Creating a database backup job
- Procedure: Running a database backup job

Procedure: Creating a database backup job

Before you create a database backup job, you must ensure that the database you want to back up has been added to OpenEdge Management.

Follow these steps to create an online or offline database backup job using a job template:

Steps	Action
1	In OpenEdge Management, click **Jobs > New > Job** from a Template. The **Create Custom Job** page appears.
2	In the **Database Maintenance** area of the **Create Custom Job** page, click: • The **OnlineBackup** link to create an online backup job. • The **OfflineBackup** link to create an offline backup job. Depending on the option you select, the **Database Backup Job** page for online or offline backup appears. The fields and options available on the Database Backup Job page are identical for both online and offline backups.
3	Specify a name for the job in the **Name** field.
4	Optionally, provide a description for the job in the **Description** field.
5	From the **Database** list, select the database you want to back up.

Steps	Action
	In the **Backup device** section, specify the backup medium. • If you select **Tape**, specify the absolute path for the tape device. • If you select **Disk**, specify the location for the backup in the Directory field and the name of the backup in the File name field. If you want to overwrite an existing backup at the specified location, select the Overwrite file option.
6	If you want to perform an incremental backup, select the **Incremental** option.
7	Optionally, specify values for various backup parameters such as volume size (-vs), redundancy factor (-red), estimate size (-estimate), compression (-com), and no recovery (-norecover).
8	Optionally, select the **Debug log file** option to generate a debug log file. Once the job is run, the debug log file becomes available through the debug log file link. The file provides diagnostic details that help you debug the job.
9	Click **Save** to save the backup job. The backup job is created and a summary of the job appears.

Procedure: Running a database backup job

After you create a database backup job, you can run it any time or schedule it to run automatically.

Follow these steps to run a database backup job:

Steps	Action
1	In OpenEdge Management, click **Jobs > Go to Jobs**. The Jobs page appears.
2	Click **View Defined Jobs** to display a list of existing jobs.
3	Select the backup job you want to run.
4	• If you want to run the backup job immediately, click **Run Now**.The job is run. • If you want to schedule the backup job, click **Schedule** to open the **Job Schedule** page. Then, specify the start date, start time, and frequency for running the job. Finally, click **Save** to save your schedule.The **Database Backup Job** page appears. The job will be run at the scheduled date and time.

4

Backing up a database using alternative methods

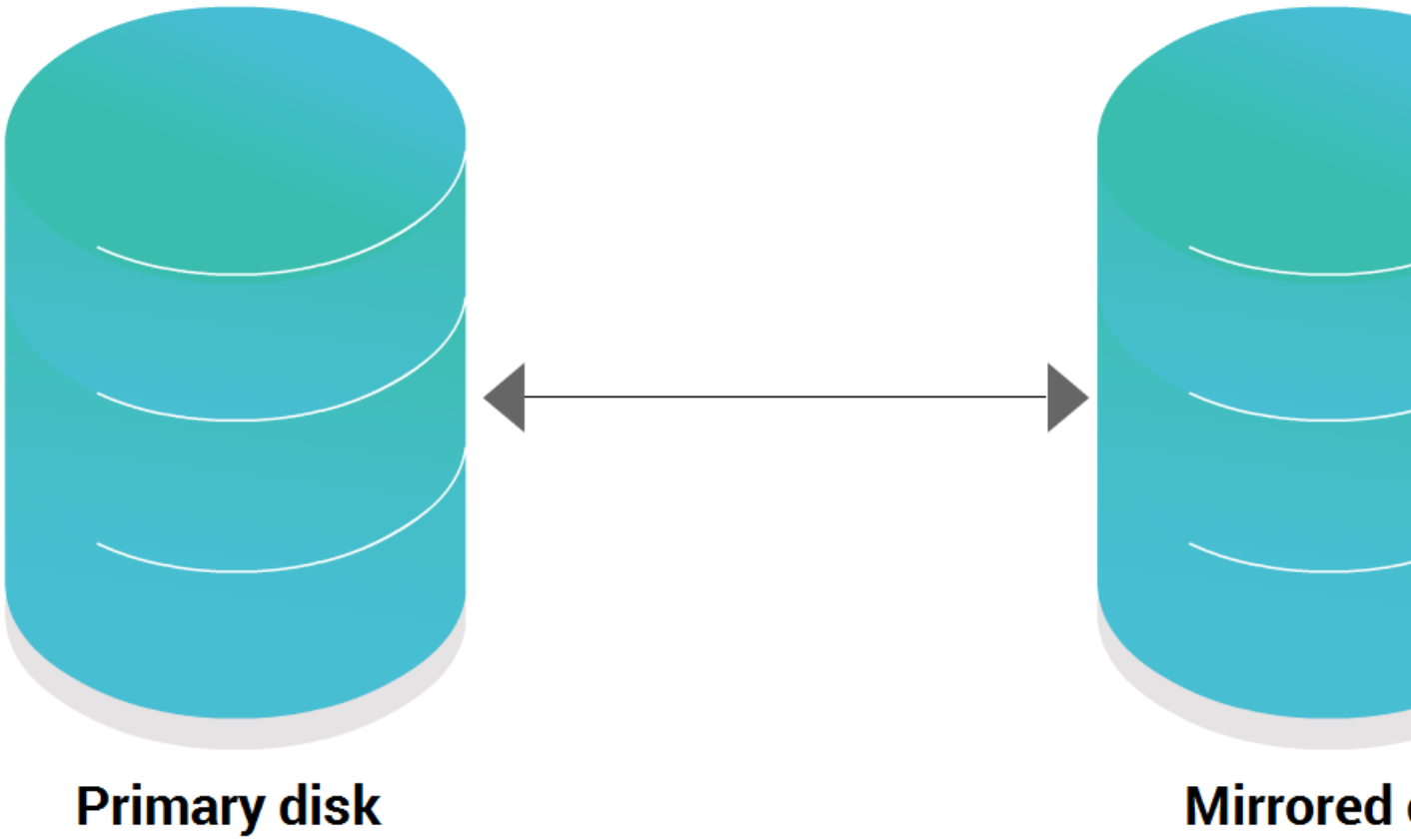
If your corporate backup and recovery strategy requires that your database backup plan use OS or third-party tools, then you can back up a database using either of the following alternative methods:

- Using OS disk mirroring online
- Using OS or third-party backup tools offline
- Next, you will learn about each of these methods.

For details, see the following topics:

- Backing up a database online using OS disk mirroring
- Backing up a database offline using OS or third-party backup tools

Backing up a database online using OS disk mirroring



If you have an OpenEdge Enterprise RDBMS license and you employ OS disk mirroring to provide data redundancy as part of your corporate backup and recovery strategy, you can perform an online full backup of a database using a combination of database quiet points. When a database quite point is enabled, all write activity to a database is stopped. The write activity resumes when the quite point is disabled. and an OS mirror fracture (split) operation.

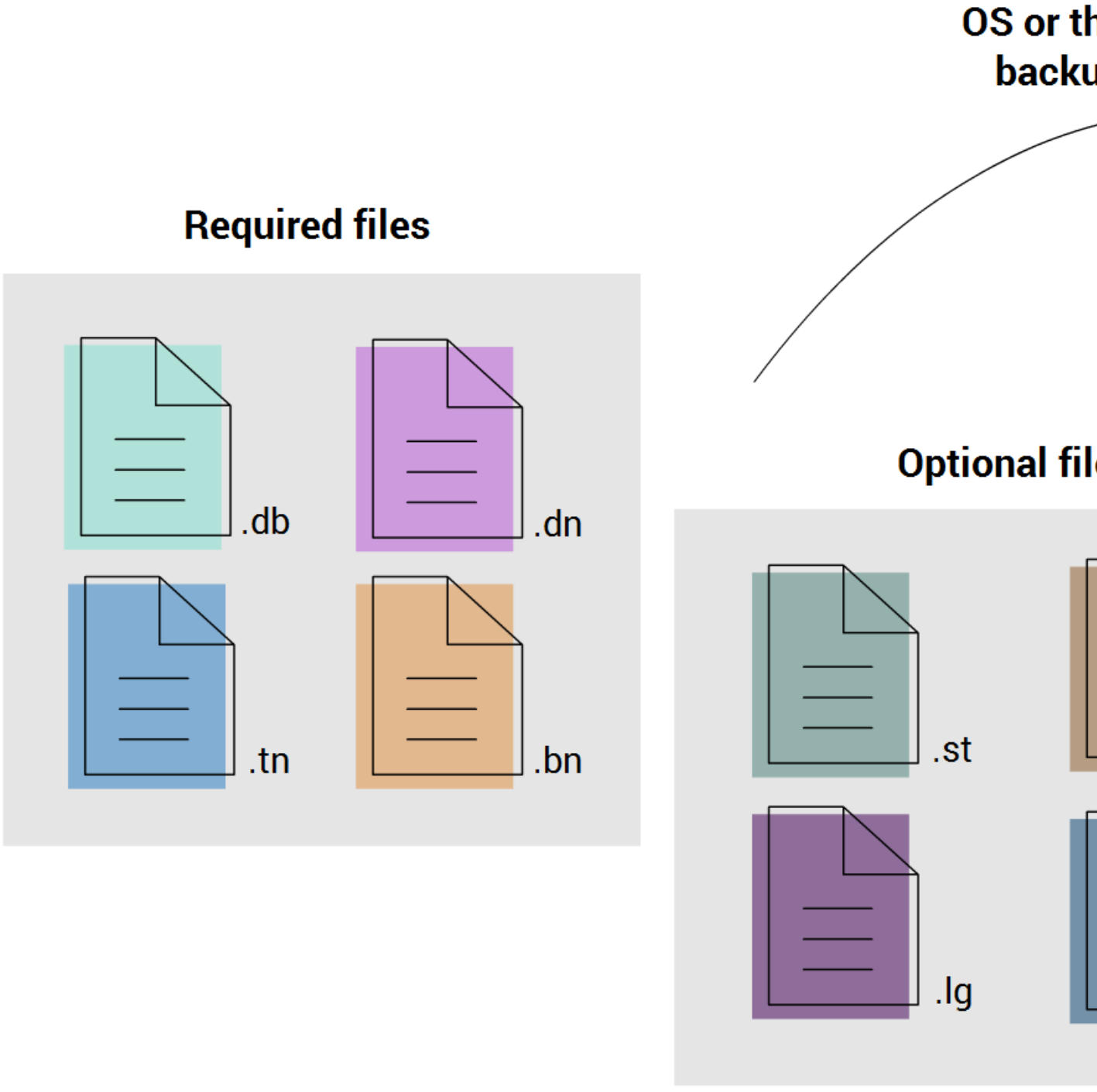
Procedure: Backing up a database online using OS disk mirroring

Follow these steps to perform an online full backup of a database using OS disk mirroring:

Step	Action
1	Enable a database quiet point on a database using PROQUIET. Use this syntax: `proquiet db-name enable` During a quiet point, write activity to the database is stopped.

Step	Action
2	Perform the OS mirror fracture (split) operation using an OS tool. Once this command is successfully executed, the fractured disk contains a duplicate of the running database (also known as the fractured version of the database). The fractured database cannot yet be started.
3	Disable the database quiet point on a database using PROQUIET. Use this syntax: `proquiet db-name disable` Write activity to the database resumes.
4	Update the .st file of the fractured version of the database by replacing the locations of the extents, which still reference the running database, with the locations of the extents of the fractured version.
5	Update the .db file of the fractured version of the database using PROSTRCT REPAIR and the .st file you updated in Step 4. The fractured version of the running database is fully functional and can be started.
6	Back up the fractured version of the database using PROBKUP. Use this syntax: `probkup db-name -norecover` **Note:** The NORECOVER parameter prevents PROBKUP from performing crash recovery or from switching to a new AI extent as part of the backup process; PROBKUP, however, does back up the BI files.

Backing up a database offline using OS or third-party backup tools



As part of your corporate backup and recovery strategy, if you back up all the files in your IT environment using OS or third-party backup tools, then you can perform an offline full backup of all the database files using those tools.

When you use OS or third-party backup tools, back up the AI files to a separate location if the database is enabled for after-imaging.

Procedure: Backing up a database offline using OS or third-party backup tools

Follow these steps to perform an offline full backup of a database using OS or third-party backup tools:

Step	Action
1	Shut down the database.
2	Back up all the database files of the types listed below to another disk or another location using OS or third-party backup tools: • Database structure (.db) file • Before-image (.bn) files • Data (.dn) files • Transaction log (.tn) files, if enabled • Structure description (.st) file • Database log (.lg) file • After-image files (.an), if enabled • Key store (.ks) file, if Transparent Data Encryption is enabled
3	Mark the database as backed up using the RFUTIL utility. Use this syntax: `rfutil db-name –C mark backedup` **Note:** RFUTIL is built into PROBKUP. If you use PROBKUP, you do not have to perform this step.

5

Verifying a database backup



After you back up a database, it is recommended that you verify the integrity of the backup. You should verify a backup immediately after generating it because if there are errors in the backup, you can redo the backup right away.

You can verify a database backup using the PROREST utility or OpenEdge Management. Note that verifying a backup works only if you backed up a database using PROREST or OpenEdge Management, not OS or third-party tools.

For details, see the following topics:

- Using PROREST to verify a database backup
- Procedure: Using OpenEdge Management to verify a database backup

Using PROREST to verify a database backup

The PROREST utility can perform two types of verification on database backups:

- Partial verification (-vp)—Checks the backup for any corrupted database blocks. You can use this parameter online or offline.
- Full verification (-vf)—Compares the backup block-for-block with the database. You use this parameter only if you have done an offline backup and your database is still offline. If changes are made to the database before you perform a full verification, then the verification will fail.

The command to verify a database backup is:

```
prorest db-name backup-filename –vp | -vf
```

Note: If you do not specify either of the above parameters in PROREST, the backup file is restored to a database.

Suppose you have backed up the sports database offline to a backup file, sports_full.bkp. Here is an example of how to perform a partial verification of the backup either online or offline:

```
prorest sports ..\backups\sports_full.bkp –vp
```

Assume that the sports database is still offline. Here is an example of how to perform a full verification of the backup offline:

```
prorest sports ..\backups\sports_full.bkp –vf
```

Procedure: Using OpenEdge Management to verify a database backup

If you have OpenEdge Management, you can use it to verify a backup.

Using OpenEdge Management, you can create a backup verification job and then run it to verify a database backup. The easiest way to do this is to use the **DatabaseRestore** job template to create a backup verification job.

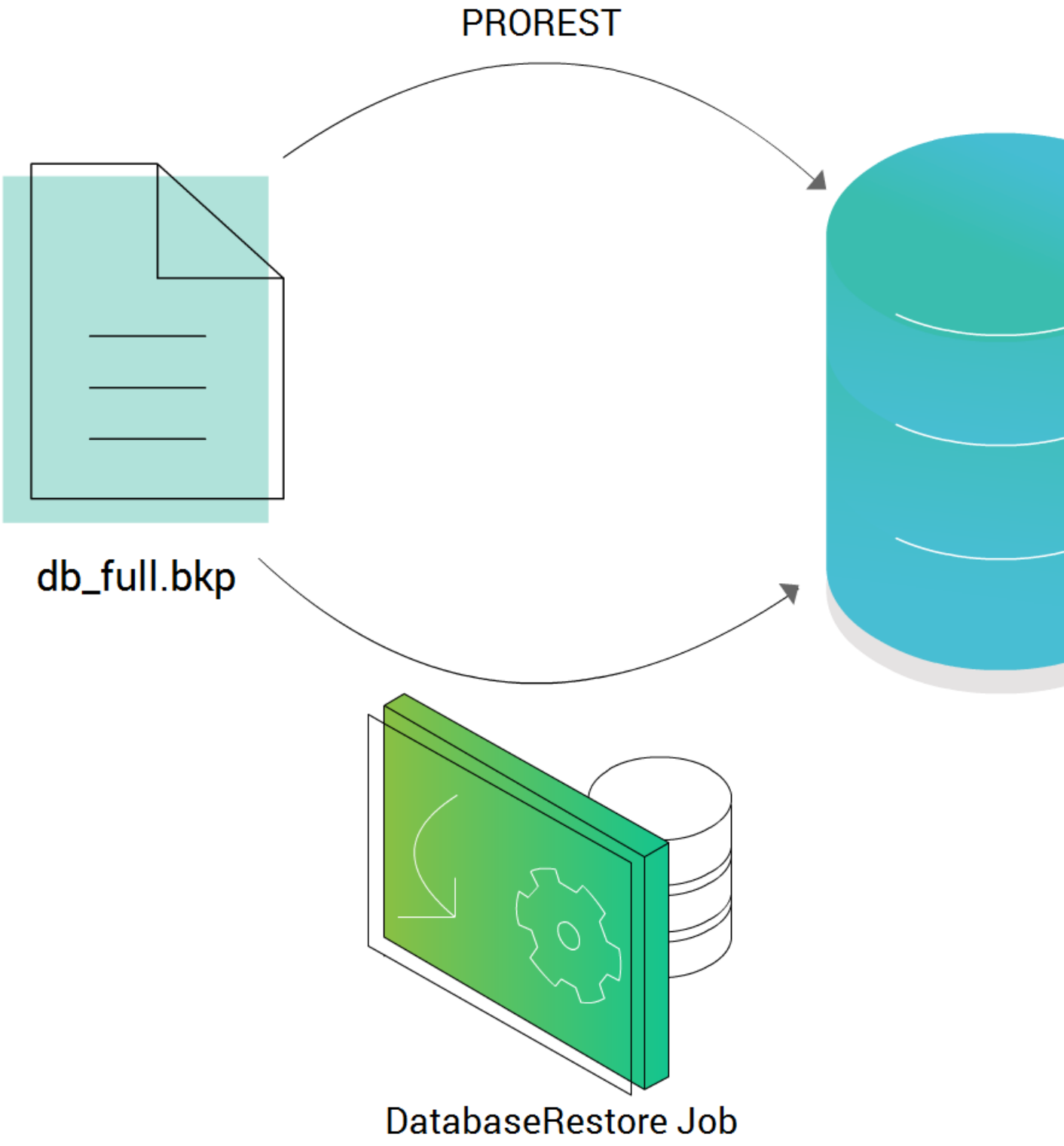
Follow these steps to create a backup verification job using the DatabaseRestore job template:

Steps	Action
1	In OpenEdge Management, click **Jobs > New > Job** from a Template. The **Create Custom Job** page appears.
2	In the **Database Maintenance** area of the **Create Custom Job** page, select **DatabaseRestore**. The **Database Restore Job** page appears.
3	Specify a name for the job in the **Name** field.
4	Optionally, provide a description for the job in the **Description** field.
5	In the **Restore** from section, specify one of the following: • If the backup file is on a tape, select **Tape** and specify the absolute path for the tape device. • If the backup file is on a disk, select **Disk** and specify the location for the backup in the **Directory** field and the name of the backup in the **File name** field.
6	In the **Restore to** section, select a container name representing the computer on which you want to restore the backup. Then, enter the full path and name of the database against which you want to verify the backup.
7	• To perform partial verification of the backup, select the **Partial verify (-vp)** option. • To perform full verification of the backup, select the **Full verify (-vf)** option.
8	Optionally, select the **Debug log** file option to generate a debug log file. Once the job is run, the debug log file becomes available through the debug log file link. The file provides diagnostic details that help you debug the job.
9	Click **Save** to save the backup verification job. The backup verification job is created and a summary of the job appears.

After you create a backup verification job, you can run it immediately or schedule it to run at a date and time of your choice. The procedure for running a backup verification job is the same as that for running a database backup job.

6

Restoring a database



In case of data loss, you can restore a database from the most recent full and incremental backups. You use the PROREST utility to restore full and incremental backups. If you have OpenEdge Management, you can use it to create a database restore job and run it whenever required.

In a production environment, you may seldom be required to restore a database. However, restoring a database is a mission-critical task. As a database administrator, you must regularly test restoring a database to verify if your backup and restore procedures work properly.

Next, you will learn about the rules for restoring a database and how to restore a database using the PROREST utility and OpenEdge Management.

For details, see the following topics:

- Rules for restoring a database
- Using PROREST to restore a database
- Procedure: Using OpenEdge Management to restore a database

Rules for restoring a database

When you restore a database, you must follow these rules:

- Restore a full backup either to a new database in the same location as the original database or to a different location.

If you restore it to a different location, you may need to modify the .st file before performing the restore.

- After restoring a full backup, do not start the database, but restore the incremental backups immediately. Starting a database after a full backup is restored changes its internal timestamps and prevents subsequent incremental backups from being applied.
- Restore a backup with the same OpenEdge RDBMS version used to perform the backup.

Restore full and incremental backups in the same order that you backed them up:

Restore a full backup first.

Restore the first incremental backup.

Restore the second incremental backup, and so on, until the restoration is completed.

- If a system failure occurs during the restore, restart the restore beginning with the backup file used at the time of the failure.

The restored database has the same database block size and storage area configuration as the original database.

Using PROREST to restore a database

You use the PROREST utility to restore a database. When you perform the restore, if you do not have a .st file named after the target database, PROREST by default restores the target database by creating one variable-length extent per storage area.

As a best practice, you should make a copy of the .st file of the source database and name it after the target database. Then edit the new .st file to reflect the extent pathnames and locations of the target database, and use that .st file in the restore process.

The command to restore a database from a full or incremental backup is:

```
prorest db-name backup-filename
```

Note: When you use PROREST to restore the backup of a large-files-enabled source database into a non-large-files-enabled target database, it will enable large files for the target database.

Suppose you have a full backup file, **sports_full.bkp**, an incremental backup file, **sports_incr.bkp**, and a .st file, **newsports.st**. To restore the full backup to a new database, **newsports**, enter:

```
prorest newsports ..\backups\sports_full.bkp
```

To restore the incremental backups to the **newsports** database, enter:

```
prorest newsports ..\backups\sports_incr.bkp
```

Generating a storage area description

If for some reason you do not have a .st file of the source database, then you can use the PROREST utility with the LIST qualifier to generate a storage area description from a backup.

The command to generate a storage area description from a backup is:

```
prorest db-name backup-filename –list
```

The output shows the size, records per block, blocks per cluster, and area number for each storage area. However, it does not provide information about the number and size of fixed and variable extents in each storage area.

You can use the available storage area information to create a .st file that meets your storage requirements, and then use the .st file to restore the target database.

Procedure: Using OpenEdge Management to restore a database

You can create a database restore job in OpenEdge Management and then run it to restore a full or incremental database backup. The best way to do this is to use the DatabaseRestore job template provided by OpenEdge Management.

Follow these steps to create a database restore job using the DatabaseRestore job template:

Steps	Action
1	In OpenEdge Management, click **Jobs > New > Job** from a Template. The **Create Custom Job** page appears.

Steps	Action
2	In the **Database Maintenance** area of the **Create Custom Job** page, select **DatabaseRestore**. The **Database Restore Job** page appears.
3	Specify a name for the job in the **Name** field.
4	Optionally, provide a description for the job in the **Description** field.
5	In the **Restore from** section, specify one of the following: • If the backup file is on a tape, select **Tape** and specify the absolute path for the tape device. • If the backup file is on a disk, select **Disk**. Then specify the location for the backup in the **Directory** field and the name of the backup in the **File name** field.
6	In the **Restore to** section, select a container name representing the computer on which you want to restore the backup. Then, enter the full path and name of the database to which you want to restore. If you want to overwrite an existing database at the location, select the **Overwrite database** option.
7	Select the **No** verification option to perform a restore.
8	Optionally, select the **Debug log file** option to generate a debug log file. Once the job is run, the debug log file becomes available through the debug log file link. The file provides diagnostic details that help you debug the job.
9	Click **Save** to save the database restore job. The database restore job is created and a summary of the job appears.

After you create a database restore job, you can choose to perform it immediately or schedule it to run later. The procedure for running a database restore job is similar to that for running a database backup job.