Progress®



Introduction to Threaded Database Server

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

Threaded Database Server Functionality and Benefits.....9

Initiate a Threaded Server Process.....9
Monitor Threaded Servers and Remote Connections.....10

Preface

Purpose

This book provides a brief introduction to the threaded OpenEdge database server and its functionality.

Audience

This book is intended for users familiar with OpenEdge database administration and who wish to learn more about threaded database server functionality.

Organization

Threaded Database Server Functionality and Benefits on page 9

Provides an overview of the functionality and benefits of the threaded database server.

Documentation conventions

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

1

Threaded Database Server Functionality and Benefits

Under the classic model, remote client requests are processed individually by the database server. Using the threaded model, remote client requests are processed concurrently. A threaded server can spawn and manage client threads up to a maximum of the value specified by the `-Ma` parameter.

Concurrent processing of remote client requests improves database performance and provides for horizontal scalability. Concurrent processing also ensures continuous availability as remote client process kills will not crash the database.

Both the threaded database server and the server-side join startup parameters are turned on by default.

For details, see the following topics:

- Initiate a Threaded Server Process
- Monitor Threaded Servers and Remote Connections

Initiate a Threaded Server Process

Initiate a threaded server process using:

```
proserve dbname -S sname -H hname -threadedServer n -threadedServerStack n
```

The following table summarizes the parameters `-threadedServer` and `-threadedServerStack`:

Parameter	Data Type	Characteristics
`-threadedServer`	Integer	Initiates the threaded database server Maximum value: 1 - Enabled Minimum value: 0 - Disabled Default value: 1
`-threadedServerStack`	Integer	Defines the thread stack size for threaded database servers in Kb. Maximum value: 2147483648 Minimum value: 64 Default value: 512

When `-threadedServer 1` is specified, the broker will spawn a threaded database server. The executable name of the threaded database server is `_mtprosrv` as opposed to `_mprosrv` for the classic database server.

Monitor Threaded Servers and Remote Connections

Threaded servers may be monitored using the Promon command and identified in the _Content VST table. Threaded servers are identified in the **Type** field of the Promon report and the _Connect VST table as **TSRV**. Remote client connections to the threaded server are displayed in the following fields:

Field name	Data type	Description
SPID	Integer	Identifies the Server Process I.D. of the remote client connection. The SPID matches the Process ID (PID) of the threaded server (TSRV).
STID	Integer	Identifies the Server Thread I.D. of the connected remote client. The STID is unique to each connected remote client.

Remote client connections may be terminated using the PROSHUT command. Released threads will be reassigned by the threaded server to new remote client connections.