Progress®



OpenEdge® Database:
Introduction to Threaded OpenEdge® Database Server

OpenEdge®

Copyright

© **2019 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.

Corticon, DataDirect (and design), DataDirect Cloud, DataDirect Connect, DataDirect Connect64, DataDirect XML Converters, DataDirect XQuery, DataRPM, Deliver More Than Expected, Icenium, Kendo UI, Kinvey, NativeScript, OpenEdge, Powered by Progress, Progress, Progress Software Developers Network, Rollbase, SequeLink, Sitefinity (and Design), Sitefinity, SpeedScript, Stylus Studio, TeamPulse, Telerik, Telerik (and Design), Test Studio, and WebSpeed 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, DataDirect Autonomous REST Connector, DataDirect Spy, SupportLink, DevCraft, Fiddler, JustAssembly, JustDecompile, JustMock, Kinvey Chat, NativeChat, NativeScript Sidekick, OpenAccess, ProDataSet, Progress Results, Progress Software, ProVision, PSE Pro, SmartBrowser, SmartComponent, SmartDataBrowser, SmartDataObjects, SmartDataView, SmartDialog, SmartFolder, SmartFrame, SmartObjects, SmartPanel, SmartQuery, SmartViewer, SmartWindow, 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.

January 2019

Last updated with new content: Release 12.0 ESAP

Updated: 2019/03/01

Table of Contents

Chapter 1: Threaded Database Server Functionality and Benefits.....7

Initiating a Threaded Server Process.....8
Monitoring Threaded Servers and Remote Connections.....8
Administering Remote Client Connection Threads with Threaded Database Server.....9

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.

On release, OpenEdge 12.0 ships with both the threaded database server and the Server-side JOIN start-up parameters turned on by default. More information is available in the OpenEdge 12.0 Database Performance Enhancements presentation found in the OpenEdge CVP Second Virtual Open House forum at https://community.progress.com/community_groups/openedge_customer_validation_program .

For details, see the following topics:

- Initiating a Threaded Server Process
- Monitoring Threaded Servers and Remote Connections
- Administering Remote Client Connection Threads with Threaded Database Server

Initiating 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.

Monitoring 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.

Administering Remote Client Connection Threads with Threaded Database Server

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