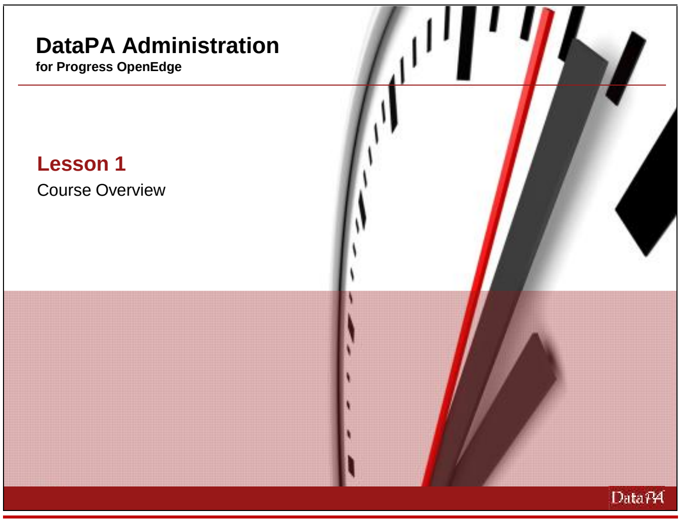
DataPA Administration

For Progress OpenEdge



DataPA Administration

for Progress OpenEdge

Lesson 1

Course Overview

Lesson 1 – Course Overview

Introduction

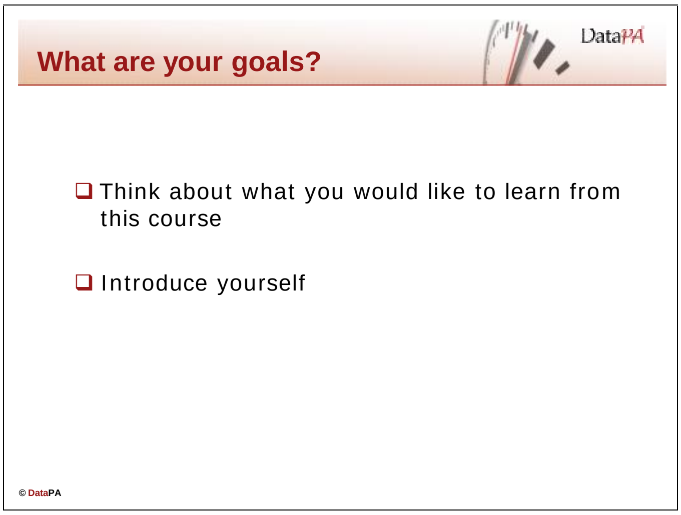
This course covers the skills and knowledge required to setup and administer DataPA for Progress OpenEdge Applications.

Audience

The course is intended for administrators who will install or maintain DataPA for Progress applications. The course will not cover Progress DataServer administration for using DataPA with non-Progress applications.

Prerequisites

Students should be familiar with Progress database administration and have a working knowledge of the Progress ABL.



What are your goals?

❑ Think about what you would like to learn from this course

❑ Introduce yourself

© DataPA

Student Goals

Please take a few moments to document your own goals for this course. What will you need to know and/or produce when you return to work?

What are the three things you most want to know about DataPA?

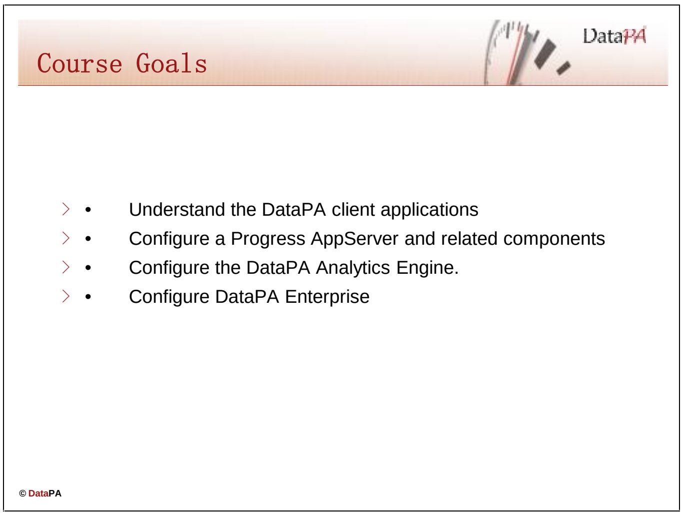
-
-
-

Please introduce yourself by answering the following questions

1. Your Name and Job

2. Your experience with Progress and DataPA

3. What you would like to learn from this course



Course Goals

- Understand the DataPA client applications
- Configure a Progress AppServer and related components
- Configure the DataPA Analytics Engine.
- Configure DataPA Enterprise

© DataPA

Course Goals

When you complete this course you should be able to:

- Understand the DataPA client applications
- Configure a Progress AppServer and related components
- Configure the DataPA Analytics Engine.
- Configure DataPA Enterprise

Lesson Overview

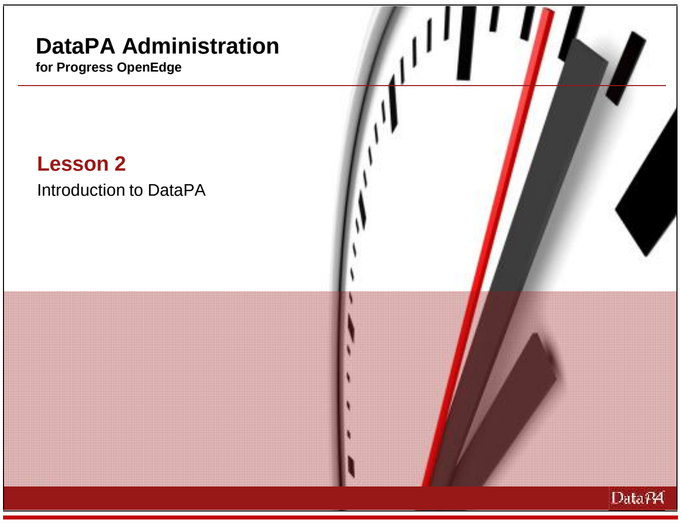
DataPA

- Lesson 1: Course Overview
- Lesson 2: Introduction to DataPA
- Lesson 3: OpenEdge AppServer Administration
- Lesson 4: Configuring the Analytics Engine
- Lesson 5: Configuring DataPA Enterprise
- Appendix A: Using the AppServer to Provide Server Security
- Appendix B: Progress AppServer Administration

© DataPA

Lesson Overview

Lesson	What is covers
Lesson 1: Course Overview	Introductory material about this course.
Lesson 2: Introduction to DataPA	Introduces the DataPA file concepts. Covers using DataPA in Excel, the Enterprise Dashboard and DataPA Reports. Briefly covers creating and modifying queries.
Lesson 3: OpenEdge AppServer Administration	Configuring, starting and managing all components of an AppServer installation for DataPA.
Lesson 4: Configuring the Analytics Engine	Creating systems links, and subjects.
Lesson 5: Configuring DataPA Enterprise	Introduction to the key components of DataPA Enterprise, and how to install and configure them to deliver business intelligence across your organisation.
Appendix A: Using the AppServer to Provide Sever Security	How to configure server-side security for DataPA to ensure users can only access information relevant to them.
Appendix B: Progress AppServer Administration	AppServer Administration using the Progress Explorer



DataPA Administration
for Progress OpenEdge

Lesson 2
Introduction to DataPA

Lesson 2 - Introduction to using DataPA

Introduction

DataPA allows users to easily access and manipulate information from your business systems. In this lesson we will explain operating system files that DataPA uses to define and store information, and introduce the interfaces that allow us to use those files to extract data into Windows applications.

Learning Objectives

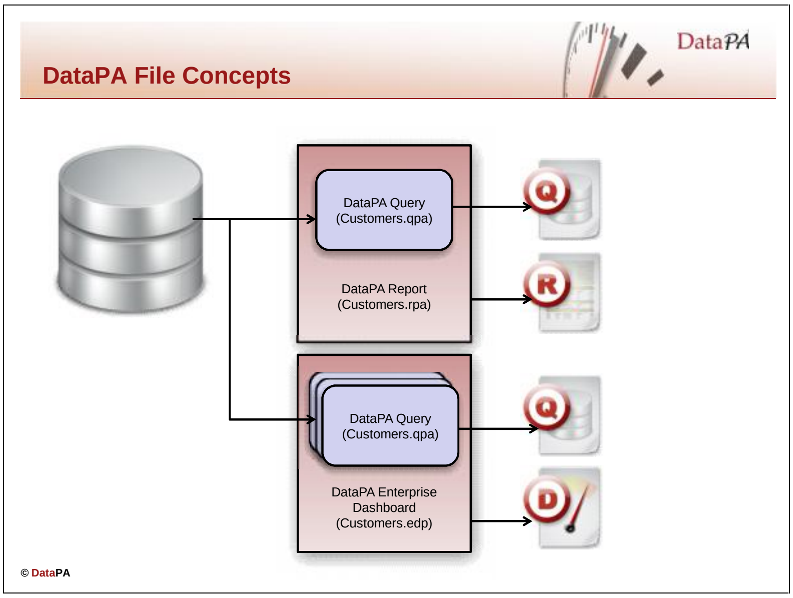
When you complete this lesson you should be able to:

- Explain DataPA file concepts.
- Use DataPA in Excel. Open and run a query.
- Use DataPA Enterprise Dashboard to open and view a dashboard.
- Use DataPA Reports. Open and run a report.
- Manipulate data in a report.
- Create and modify queries.

Prerequisites

Before you begin this lesson you should be able to:

- Open a copy of Excel



DataPA File Concepts

DataPA

DataPA Query (Customers.qpa)

DataPA Report (Customers.rpa)

DataPA Query (Customers.qpa)

DataPA Enterprise Dashboard (Customers.edp)

© DataPA

DataPA Queries

In order to define the data extracts we will use from a given system, DataPA uses the concept of a query. The query defines the following:

- The system we wish to extract data from
- The type of data we want to extract
- Any conditions or parameters used to limit the data we receive
- Which fields from the data set we wish to receive
- The sort order the data will be received in

Queries are stored on disk in the form of qpa files, for example we might save a file as C:\MyQueries\customer.qpa.

DataPA Dashboards

A dashboard can potentially contain many queries. In addition to the data extract information defined by these queries, we need to store the dashboard layout, formatting and behaviour definitions. As such, DataPA dashboards are stored in a different file format, edp files. For example we might save our customer dashboard as C:\MyReports\customers.edp. This dashboard file contains both the query, and layout, formatting and behaviour information for the dashboard.

DataPA Reports

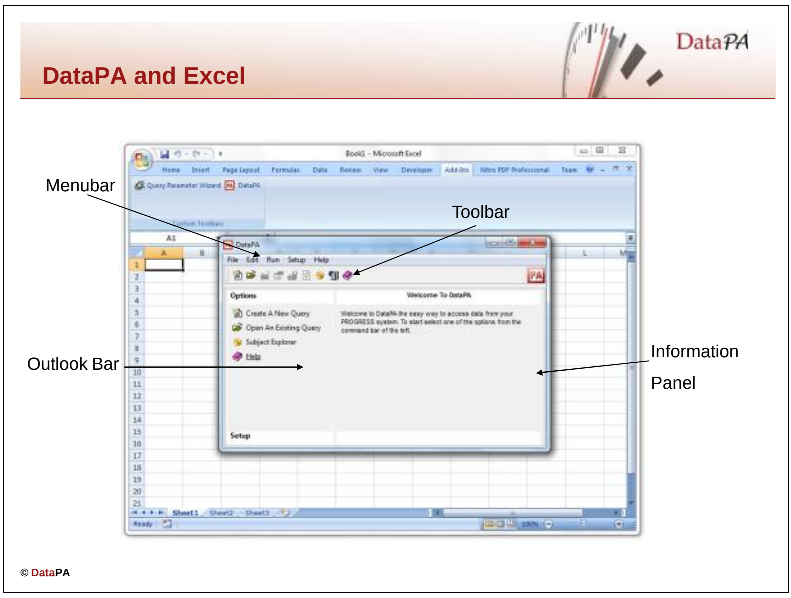
Like dashboards, in a report we need to store the data formatting and layout information in addition to the data extract definition. As such, DataPA reports are stored in a different file format, rpa files. For example we might save our customer report as C:\MyReports\customers.rpa. This report file contains both the query, and formatting and layout information for the report.

Sharing Queries

We will see later in this course how we can share queries between applications such as Microsoft Excel, DataPA Enterprise Dashboard and DataPA Reports.

You can create reports, and add queries to a dashboard by copying an existing query. Any subsequent changes to the query in the dashboard or report do not affect the existing query, as the changes are made to a copy of the query stored independently in the dashboard or report.

Similarly, you can share a query from an existing dashboard or report, and use it in Excel or a different dashboard or report. The shared query is a copy of the original, and as such any changes to it do not affect the original query stored in the original dashboard or report.



DataPA and Excel

© DataPA

Opening the DataPA Excel Add-In

When DataPA is installed it adds a button to the Excel toolbars to open the DataPA Add In. If the button is missing, follow these steps:

1. From the Excel tools menu, select Add Ins
2. Make sure DataPA is checked
3. Press OK

If DataPA is not available in the Add Ins Dialog box, follow these steps:

1. From the Excel Add Ins Dialog box, press *Browse*
2. Browse to C:\Program Files\DataPA\Excel
3. Select DataPA.xla
4. Follow the instructions above from step 2

To open DataPA in Excel, press the DataPA button on the toolbar.

Accessing the Online Help

DataPA includes a context sensitive help system.
To access help from the main screen, select *Contents* from the help menu.
There are three ways to locate information in help; these correspond to the three tabs in the Help Topics window.

Tab	Description
Contents	Expandable table of contents that allows you to search through topics in a hierarchical arrangement.
Index	An alphabetic listing of topics. You can either browse to a topic or enter a keyword to search.
Find	Uses a keyword-searchable database to find topics.

You can also access help topics relevant to the screen you are currently focused on by pressing the F1 key.

Opening a Query in Excel

To open a query, follow these steps:

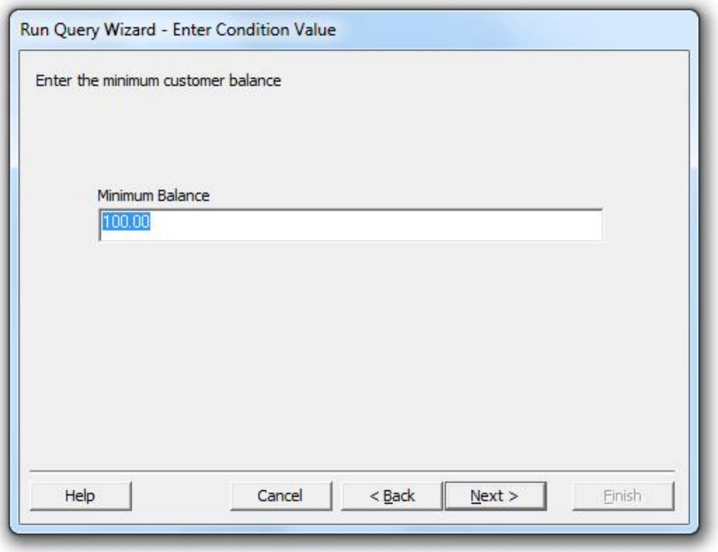
- From the *File* menu, select *Open an Existing Query*
- Select the query file in the Open File dialog and press ok.

Notice the query name and description now appear in the Information panel.

Running a Query in Excel

To run a query, follow these steps:

- From the *Run* menu, select *Run Query*
- Select *Next* to move past the introduction screen.
- Enter appropriate parameter values and press *Next*

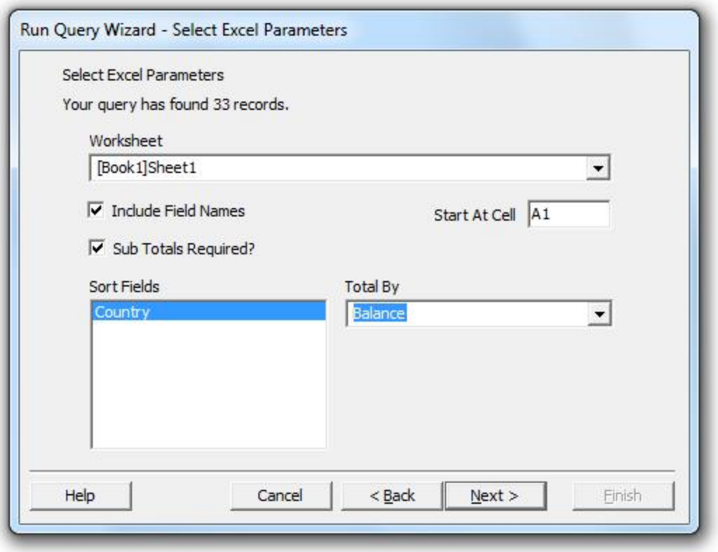


Run Query Wizard - Enter Condition Value

Enter the minimum customer balance

Minimum Balance: 100.00

- Continue pressing next to run the query until you reach the *Select Excel Parameters* screen.



Run Query Wizard - Select Excel Parameters

Select Excel Parameters

Your query has found 33 records.

Worksheet: [Book1]Sheet1

☑ Include Field Names

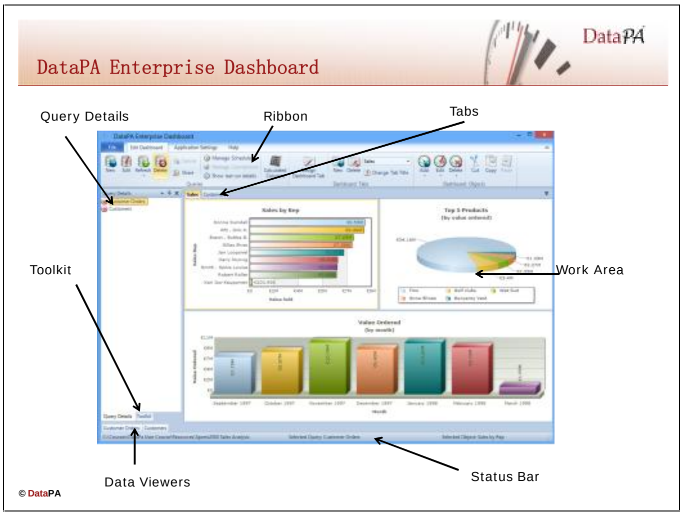
Start At Cell: A1

☑ Sub Totals Required?

Sort Fields: Country

Total By: Balance

- Select the workbook you require, the starting cell and any sub-total options.
- Press *Next* then *Finish* to complete the wizard.



Opening a Dashboard

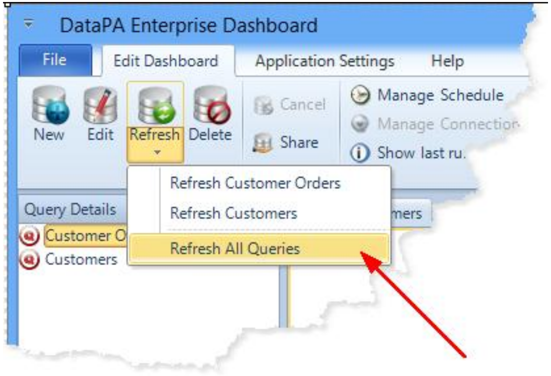
To open the DataPA Enterprise Dashboard, click on the DataPA Enterprise Dashboard icon from the start menu or on the desktop.
To open a dashboard, follow these steps:

- Select *Open an Existing Dashboard* from the *File* menu of DataPA Enterprise Dashboard.
- Select the dashboard file you wish to open and press ok.

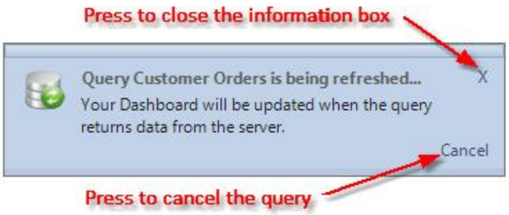
Refreshing data in a Dashboard

To refresh the data in an open dashboard in the DataPA Enterprise Dashboard, follow these steps:

- Press the left mouse button over the arrow below the *Refresh* button in the *Edit Dashboard* tab of the ribbon.
- Select the individual query you wish to refresh, or *Refresh All Queries* if you wish to refresh all queries.

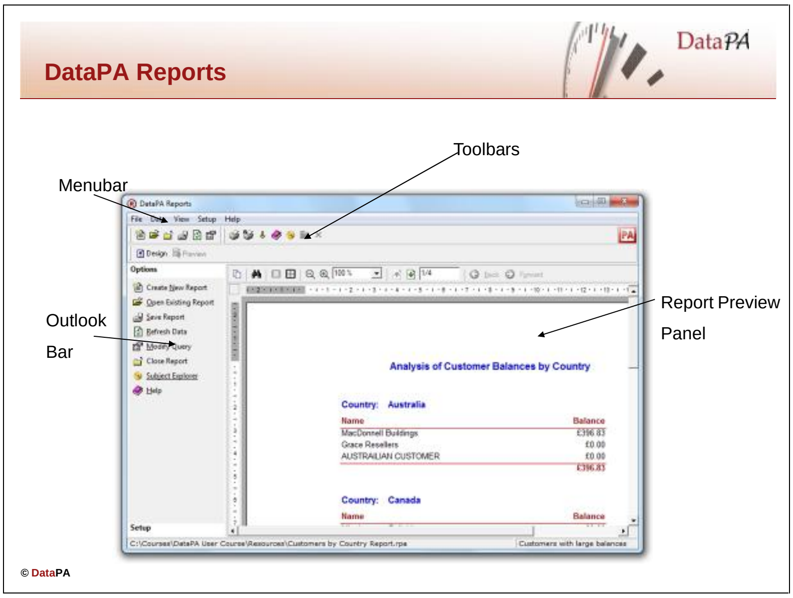


Whilst a query is being refreshed, DataPA Enterprise Dashboard displays an information box. You can cancel the query by pressing the *cancel* button on this information box.


Press to close the information box

Query Customer Orders is being refreshed...
Your Dashboard will be updated when the query returns data from the server.

Press to cancel the query


DataPA Reports

Opening a Report

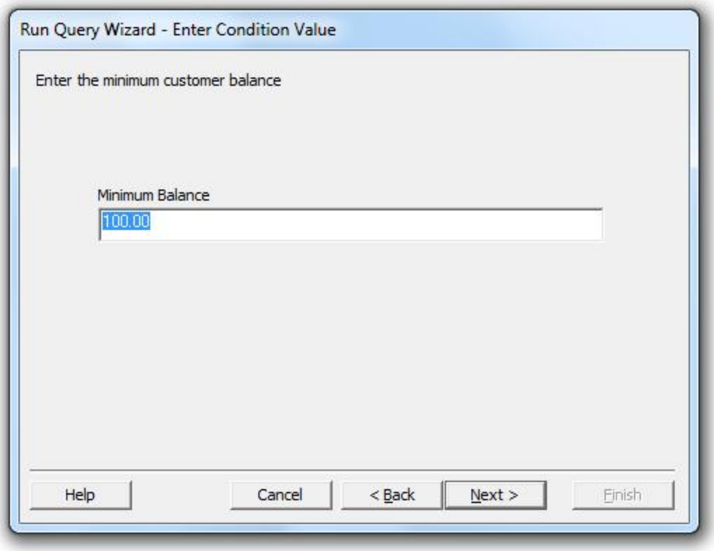
To open the DataPA Report Designer, click on the DataPA Reports icon from the start menu or on the desktop. To open a report, follow these steps:

- Select *Open an Existing Report* from the *File* menu of the report designer.
- Select the report file you wish to open and press ok.

Running a Report

To run an open report in the DataPA Report Designer, follow these steps:

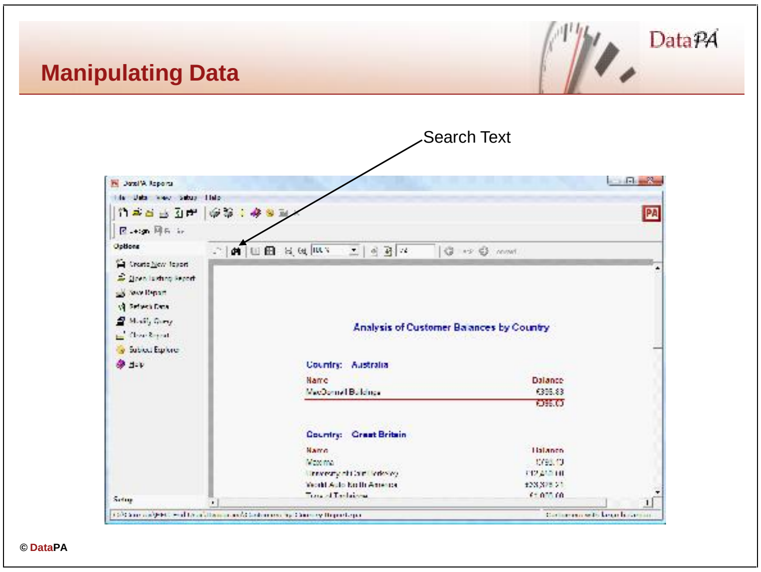
- Select *Refresh Data* from the *Data* menu of the report designer.
- Select *Next* to move past the introduction screen.
- Enter appropriate parameter values and press *Next*


Run Query Wizard - Enter Condition Value

Enter the minimum customer balance

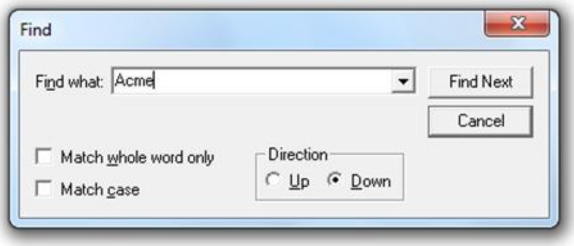
Minimum Balance: 100.00

- Press *Next* to retrieve the data for your report.



Search for data in a report.

The Report Preview panel in the DataPA Reports provide a way for searching for data within a report.

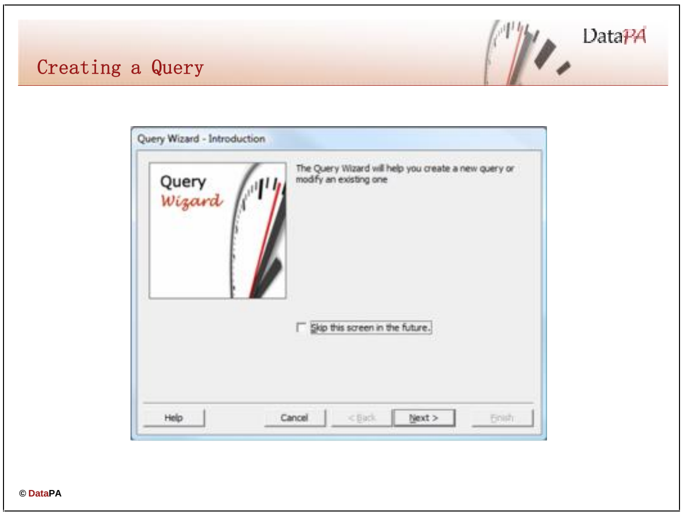
Method	Description
Search Text	Provides a simple dialog box to search for a text string in the report.  Find what: Acme ☐ Match whole word only ☐ Match case Direction ○ Up ● Down Follow these steps to search: 1. Press the *Search Text* button to open the search dialog box. 2. Enter the text you wish to search for. 3. Press *Next* to find the next instance of the text you are searching for.

Using the Multiple Page Viewer

The DataPA Reports Preview panel allows you view several pages of the report at once.

Follow these steps to view multiple pages:

1. Click the Multiple Pages button on the top of the DataPA Reports Preview panel. The icon for this is ⊞
2. Drag the mouse to highlight the number of pages you want to have in the viewer.
3. Release the mouse and view the report.



Creating a Query

Query Wizard - Introduction

The Query Wizard will help you create a new query or modify an existing one

□ Skip this screen in the future.

© DataPA

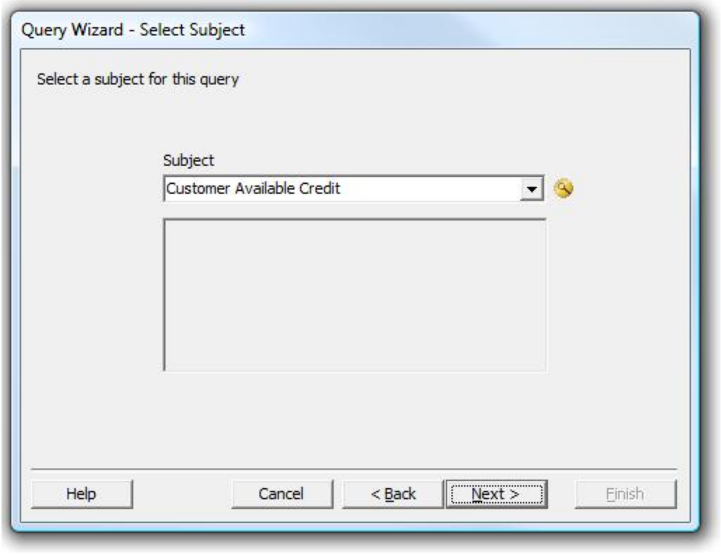
Creating a query with the query Wizard

To open the query wizard to create a new query follow these steps:

1. Open DataPA in Excel
2. From the *File* menu, select *Create a New Query*

Follow these steps to create a query using the query wizard and save it to disk:

1. Press *Next* to step past the introduction screen.
2. Enter a name and description for the query and press *Next*.
3. Select the subject you wish to create a query from and press *Next*.

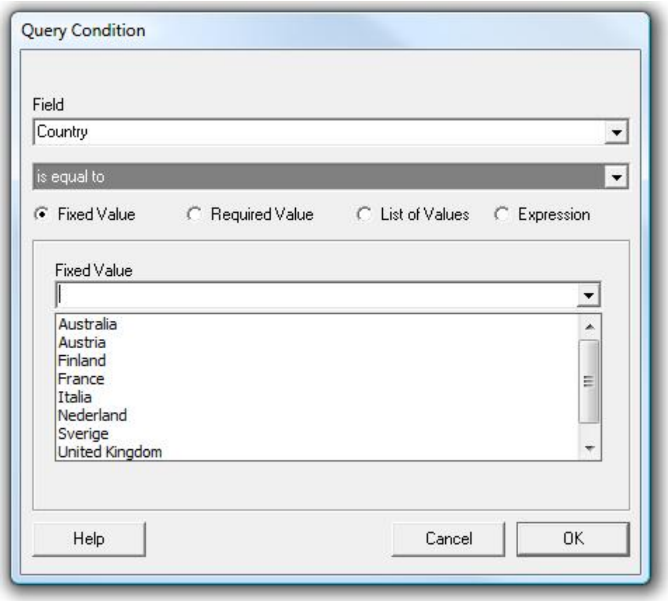


Query Wizard - Select Subject

Select a subject for this query

Subject: Customer Available Credit

4. Enter any conditions required using the *Query Condition* dialog box.



Query Condition

Field: Country

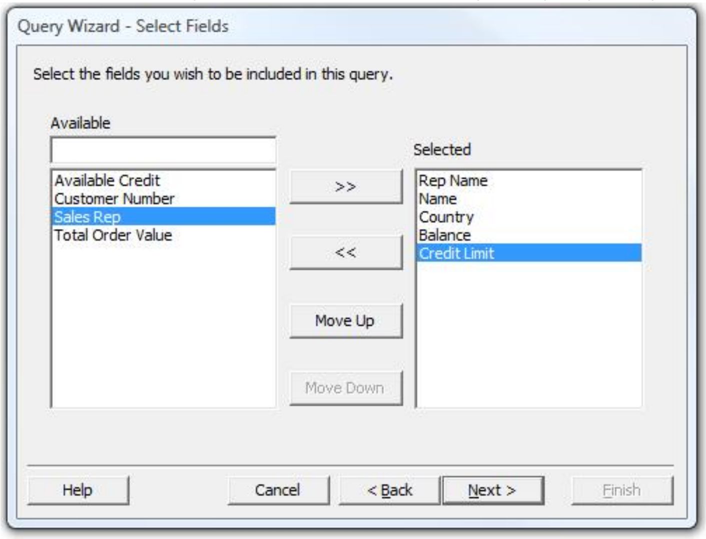
is equal to

● Fixed Value ○ Required Value ○ List of Values ○ Expression

Fixed Value

Australia
Austria
Finland
France
Italia
Nederland
Sverige
United Kingdom

5. Select the field(s) you wish to be retrieved by the query and press *Next*.



Query Wizard - Select Fields

Select the fields you wish to be included in this query.

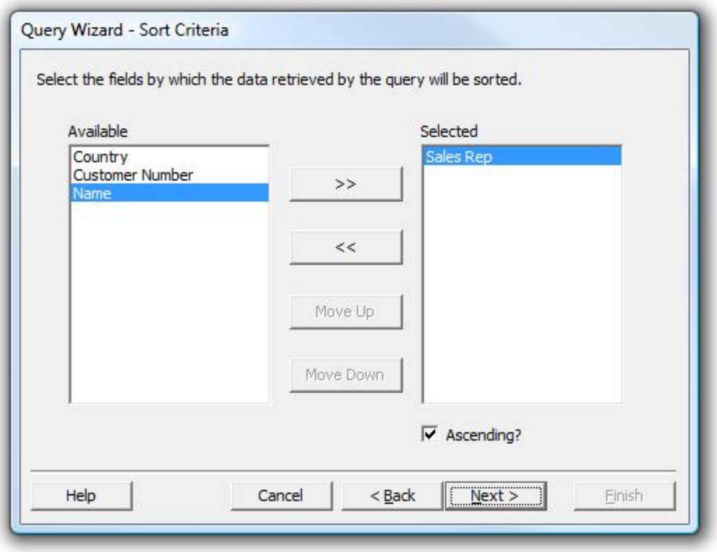
Available

Available Credit
Customer Number
Sales Rep
Total Order Value

Selected

Rep Name
Name
Country
Balance
Credit Limit

6. Select the field(s) you wish to sort the query by and press *Next*.



Query Wizard - Sort Criteria

Select the fields by which the data retrieved by the query will be sorted.

Available

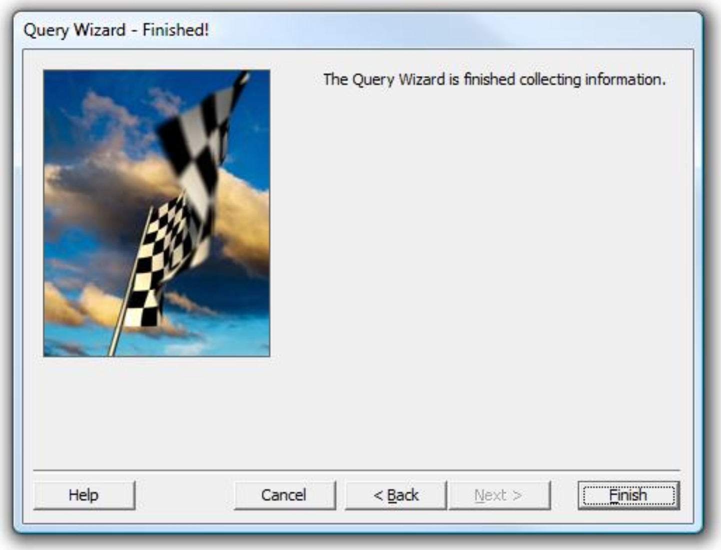
Country
Customer Number
Name

Selected

Sales Rep

☑ Ascending?

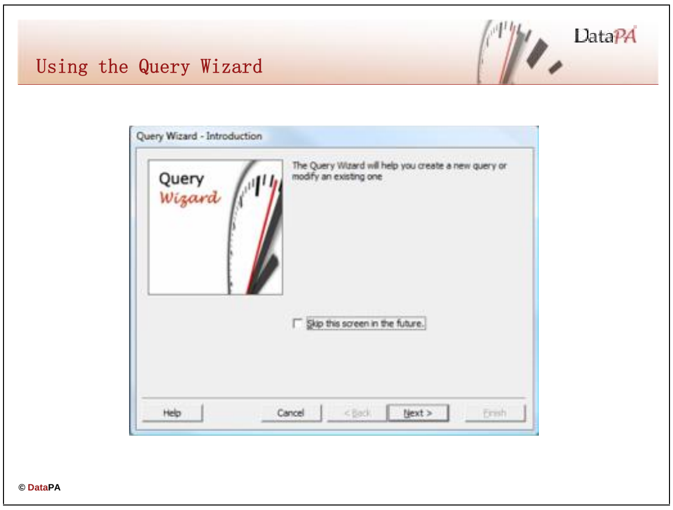
7. Press *Next* to bypass the *Add Multiple Subjects to Query* screen.
8. Press *Finish* to create your query.



Query Wizard - Finished!

The Query Wizard is finished collecting information.

9. From the *File* menu select *Save Query*.
10. Browse to the directory you wish to save your query to and press *Save*.



Using the Query Wizard

Query Wizard - Introduction

The Query Wizard will help you create a new query or modify an existing one

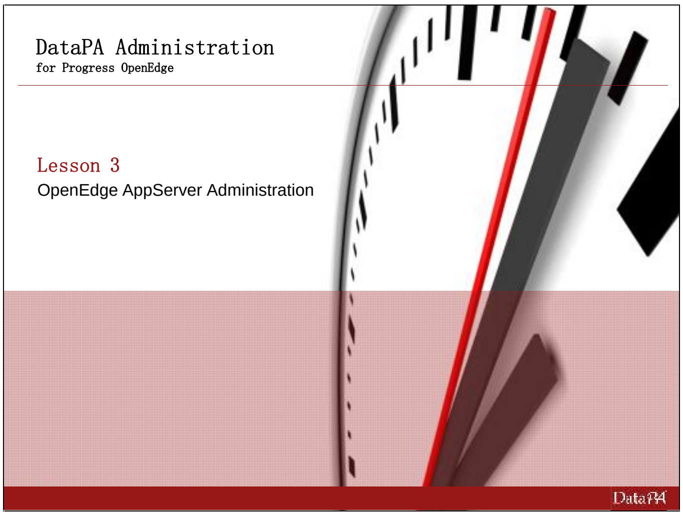
□ Skip this screen in the future.

Modifying a query with the query Wizard

To start the query wizard to modify an existing query follow these steps:

1. Open DataPA in Excel
2. Open the query you wish to modify in DataPA
3. From the *Edit* menu, select *Modify Query*

Follow the Query Wizard steps as you would to create a new query to modify your existing query.



DataPA Administration
for Progress OpenEdge

Lesson 3
OpenEdge AppServer Administration

Lesson 3: OpenEdge AppServer Administration

Introduction

This lesson shows how to configure and start AppServer components, and how to check their status while they are running. The OpenEdge AppServer allows DataPA to report on complex distributed applications by calling remote procedures from within the DataPA client.
Appendix 3 Contains the Actions required for utilising the Progress Explorer for Progress/OpenEdge Version 10.x and earlier.

Leaning Objectives

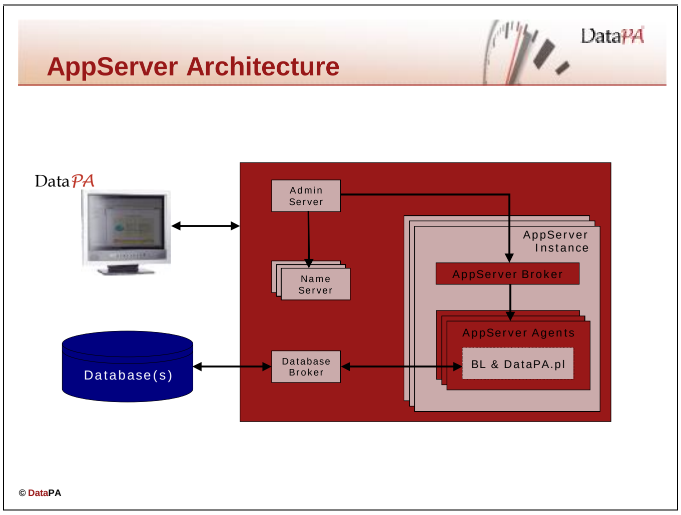
When you complete this lesson, you should be able to:

- Add and configure AppServer components
  - AdminServer
  - NameServer
  - AppServer
- Start AppServer components using the OpenEdge Explorer and check their status. Start the:
  - AdminServer
  - NameServer
  - AppServer
- Work with the ubroker.properties file and recognize its entries.
- Use command line utilities as an alternative to the OpenEdge Explorer to start, stop, and query the NameServer and AppServer broker.

Prerequisites

Before you begin this lesson you should be able to:

- Administer a OpenEdge database
- Start and stop a OpenEdge Database broker.



AppServer Architecture

© DataPA

Introduction

Before you can connect DataPA to an AppServer, you must be sure that the following components are running on the AppServer host:

AdminServer

Database broker (on the database server host)

NameServer

AppServer

Some or all of these may be set to autostart. Manual start and autostart are described later in this section.

AdminServer

The AdminServer manages the other OpenEdge components. It also allows you to configure and manage components using the Progress Explorer tool or command-line utilities.

NameServer

The NameServer:

- Maintains a list of available AppServers and the application services they support.
- Directs a DataPA connection request to an AppServer based on the requested Application Service Name.
- Balances DataPA workload among AppServers.

Database broker

The database broker manages database servers. This course will not cover database brokers.

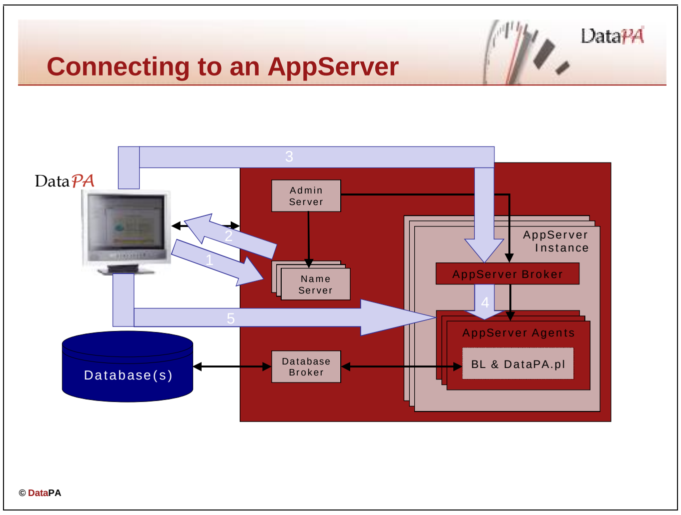
AppServer

The AppServer has two components:

AppServer Broker.

- Manages connections between DataPA clients and a pool of AppServer Agents.
- Maintains a pool of AppServer Agents.
- Routes DataPA client requests to an available AppServer Agent.
- Starts up AppServer Agents as needed to service multiple DataPA clients.
- Trims AppServer Agents when demand is low.

AppServer agents execute DataPA client requests on the server.



Connecting to an AppServer

The connection process

Progress performs several operations using various components when DataPA connects to an AppServer. Generally, the following steps occur. The details may vary somewhat if DataPA is set to connect directly to the AppServer not via the NameServer.

Step	Description
1.	DataPA contacts the NameServer on the AppServer host with the name of an Application Service that DataPA would like to use.
2.	The NameServer returns the information DataPA needs to connect with an AppServer instance that supports the requested Application Service.
3.	DataPA establishes a connection with the AppServer Broker.
4.	The AppServer Broker establishes a connection to an appropriate AppServer Agent.
5.	The broker routes requests to AppServer agents.

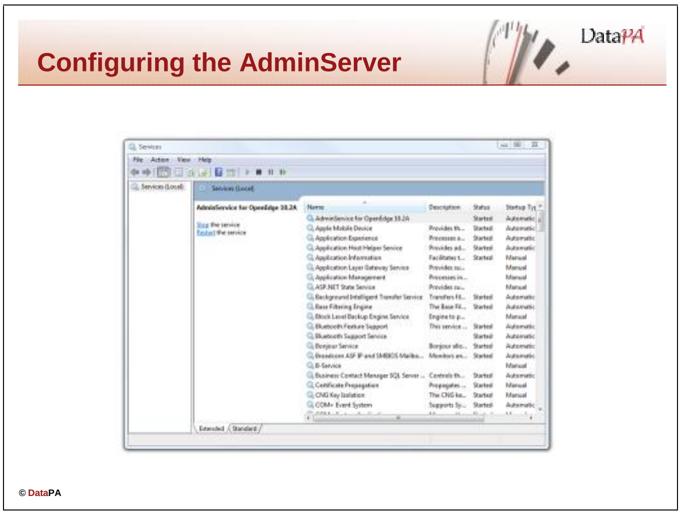
These steps indicate why specific AppServer components must be running before DataPA can connect to the AppServer.

AppServer instances

AppServer instances include an AppServer Broker and a pool of AppServer Agents. The AppServer Agents run on the AppServer host and execute ABL procedures in response to DataPA requests. The AppServer Broker manages the DataPA connections and dispatches requests to AppServer Agents.

Application Service name

An *Application Service Name* is an alias for the AppServer instance that provides a specific business function. For example, an AppServer instance might provide access to the Order Table. You might therefore name the Application Service alias “Orders”. Each instance is defined as a connection within a system in DataPA, and DataPA establishes a connection to the AppServer instance’s AppServer Broker when a query is run against that system. The broker establishes a connection to an AppServer Agent.



Configuring the AdminServer

AdminService for OpenEdge 10.2A

Stop the service
Restart the service

Name	Description	Status	Startup Typ
AdminService for OpenEdge 10.2A		Started	Automatic
Apple Mobile Device	Provides th...	Started	Automatic
Application Experience	Processes a...	Started	Automatic
Application Host Helper Service	Provides ad...	Started	Automatic
Application Information	Facilitates t...	Started	Manual
Application Layer Gateway Service	Provides su...		Manual
Application Management	Processes in...		Manual
ASP.NET State Service	Provides su...		Manual
Background Intelligent Transfer Service	Transfers fil...	Started	Automatic
Base Filtering Engine	The Base Fil...	Started	Automatic
Block Level Backup Engine Service	Engine to p...		Manual
Bluetooth Feature Support	This service ...	Started	Automatic
Bluetooth Support Service		Started	Automatic
Bonjour Service	Bonjour allo...	Started	Automatic
Broadcom ASF IP and SMBIOS Mailbo...	Monitors an...	Started	Automatic
B-Service			Manual
Business Contact Manager SQL Server ...	Controls th...	Started	Automatic
Certificate Propagation	Propagates ...	Started	Manual
CNG Key Isolation	The CNG ke...	Started	Manual
COM+ Event System	Supports Sy...	Started	Automatic

© DataPA

AdminServer on Windows

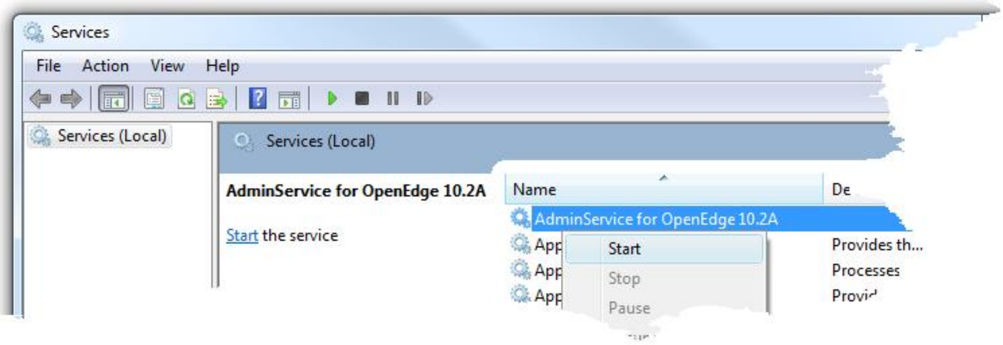
The AdminServer must be running before you start Progress processes on a host machine. The AdminServer defaults to automatically start when the system starts if installed on Windows machines. Follow these steps to see if it is running on your Windows system:

Choose Start → Settings → Control Panel → Administrative Tools → Services

Manually starting the AdminServer

To start the AdminServer:

1. Select the AdminService for OpenEdge.
2. Choose Action→Start.



AdminService for OpenEdge 10.2A

Start the service

AdminService for OpenEdge 10.2A
App Provides th...
App Processes
App Provi

Stopping the AdminServer

To stop the AdminServer on Windows:

1. Select the AdminService for OpenEdge.
2. Choose Action→Stop.

UNIX, Linux and Windows (Character or command line mode)

Use the PROADSV utility to start and stop the AdminServer on a UNIX or Linux system or from the command line on a Windows system.

PROADSV has the following syntax:

```
proadsv { { { -start { [ -adminport port-number ] }
| -stop | -query } [ -port port-number ] } | -help }
```

start – Starts the AdminServer.

adminport *port-number*–- Specifies the port number used by the AdminServer for database broker communication. If a port number is not specified, the adminport defaults to port 7832.

stop – Stops the AdminServer.

query – Displays AdminServer status.

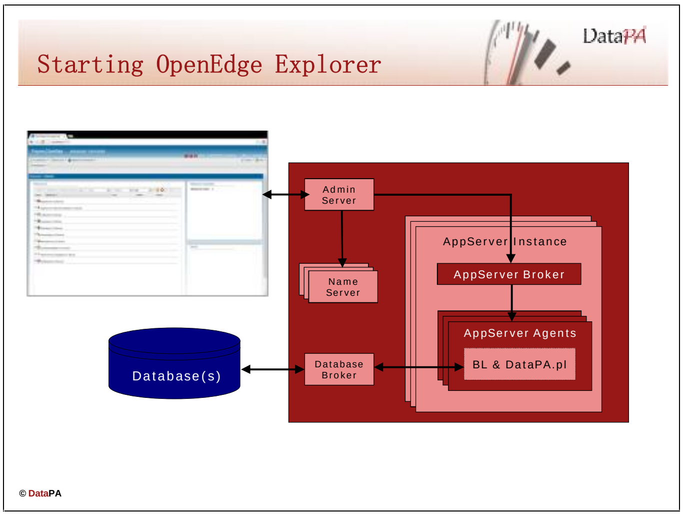
port *port-number* – Specifies the listening port number for online command utilities, such as DBMAN. If a port number is not specified, it defaults to 20931.

To check whether the AdminServer is running on UNIX systems, run the ps command to show the full command line for each process on the system and locate any jre commands in the list. The AdminServer process is running if you see a `jre` command with the arguments that correspond to those specified for `jvmstart` in the Progress `proadsv` shell script located in `OpenEdgeInstallDir/bin`.

PROADSV does not change the windows registry and so the settings you establish are not permanent.

Admserv.log file

The AdminServer writes information to a log file, `admserv.log`, in the Progress working directory (OpenEdge\Wrk by default). The log file keeps a record of AdminServer events and actions, such as starting and stopping AppServer components, and can be a useful troubleshooting aid. You can open `admserv.log` in any text editor.



Starting OpenEdge Explorer

© DataPA

What is the OpenEdge Explorer?

The OpenEdge Explorer is a web based tool that you use to configure, start, and stop, and retrieve status for NameServers, AppServers, and database connections. The OpenEdge Explorer reads and writes configuration information into files named `conmgr.properties` for databases and `ubroker.properties` for Appservers and Nameservers. Ubroker.properties will be described in more detail later in this lesson. The OpenEgde Explorer works with the AdminServer on the AppServer host to manage AppServers and NameServers.

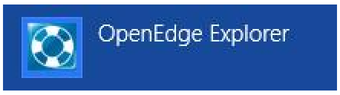
The OpenEdge Explorer and the AdminServer

The OpenEdge Explorer tool works with the AdminServer on the AppServer host. Therefore, the OpenEdge Explorer must connect to a running AdminServer before you can use it to manage the AppServer components.

Starting the OpenEdge Explorer

Follow these steps to start the Progress Explorer:

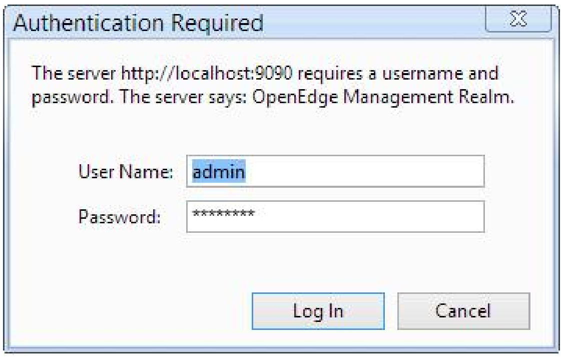
1. Choose Start→Programs→OpenEdge→Progress Explorer tool. (The Progress Explorer Tool is on the OpenEdge menu):



2. Type "localhost:9090" into a Browser URL.

Note: If your AdminServer does not use the default port, possibly because you run multiple AdminServers for different versions of OpenEdge, you can change the AdminServer Connection port number on the browser URL

You may, if required, have to enter a username and password to access the OpenEdge Explorer



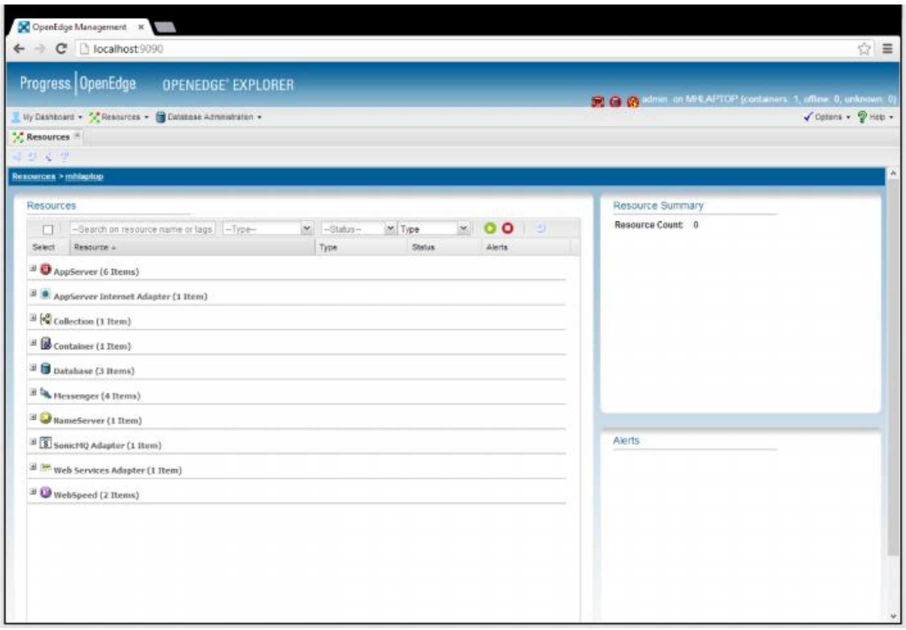
Authentication Required

The server http://localhost:9090 requires a username and password. The server says: OpenEdge Management Realm.

User Name: admin

Password: ********

The OpenEdge Explorer Tool is shown below:



localhost:9090

admin on MHLAPTOP (containers: 1, offline: 0, unknown: 0)

Resources > mhlaptop

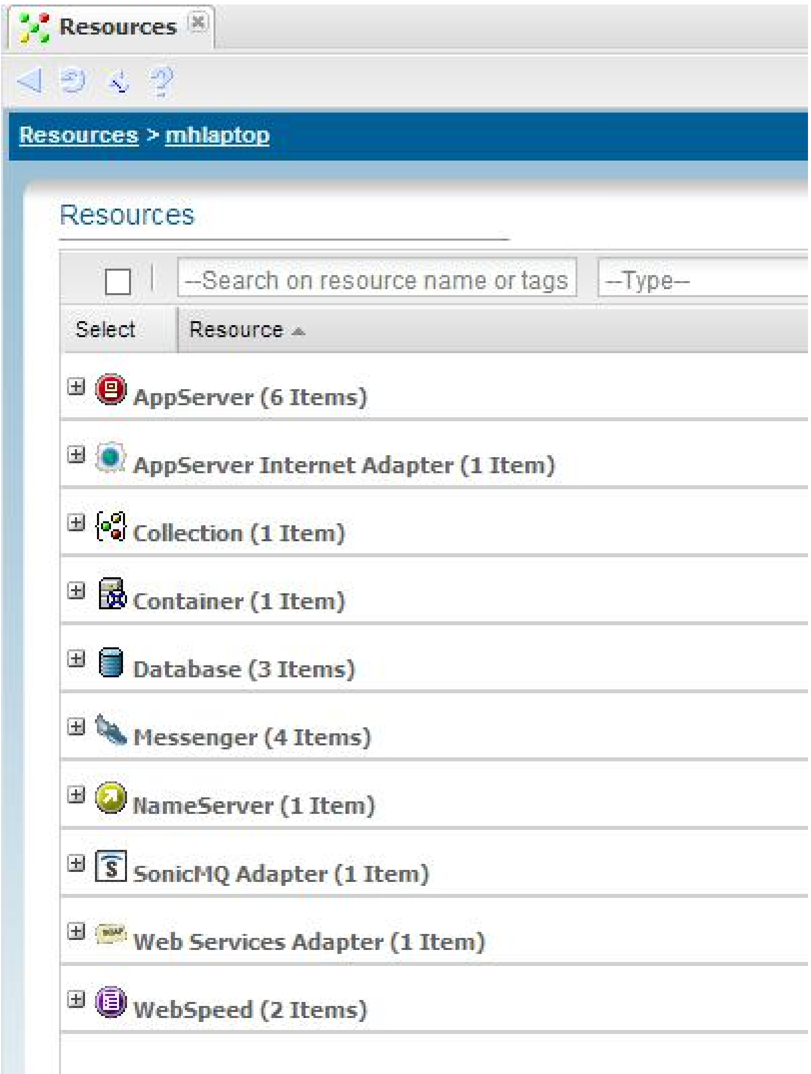
Resource Count: 0

Select	Resource	Type	Status	Alerts
⊞	AppServer (6 Items)			
⊞	AppServer Internet Adapter (1 Item)			
⊞	Collection (1 Item)			
⊞	Container (1 Item)			
⊞	Database (3 Items)			
⊞	Messenger (4 Items)			
⊞	NameServer (1 Item)			
⊞	SonicMQ Adapter (1 Item)			
⊞	Web Services Adapter (1 Item)			
⊞	WebSpeed (2 Items)			

Stopping the OpenEdge Explorer

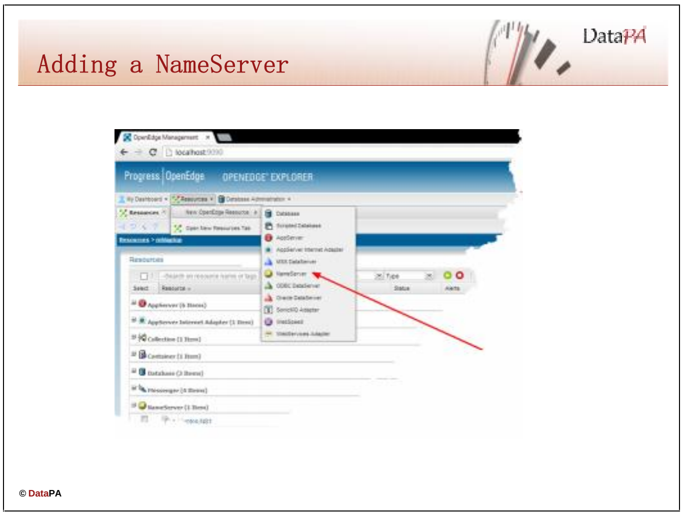
To stop the OpenEdge Explorer:
Close the Browser or Browser Tab.

After connecting, the Progress Explorer window should resemble the following:



- AppServer (6 Items)
- AppServer Internet Adapter (1 Item)
- Collection (1 Item)
- Container (1 Item)
- Database (3 Items)
- Messenger (4 Items)
- NameServer (1 Item)
- SonicMQ Adapter (1 Item)
- Web Services Adapter (1 Item)
- WebSpeed (2 Items)

The OpenEdge Explorer window now includes folders for a number of system components, including AppServer components (Database connections, AppServer, NameServer).


Adding a NameServer

DataPA

© DataPA

Introduction

The NameServer directs DataPA client requests to connect to an appropriate AppServer instance. You can have multiple NameServers, providing one level of fault-tolerance and load-balancing. See OpenEdge documentation for further details.

OpenEdge provides one NameServer, NS1, by default. It is recommended that you create your own NameServers, to avoid problems if NS1 changes with OpenEdge updates.

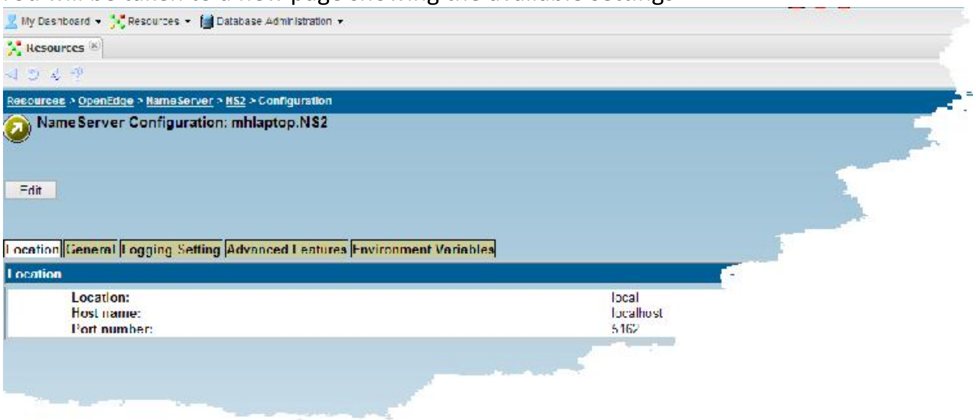
NameServer log files

As with the other components, the NameServer creates a log file. During configuration, you can choose where to store the file, how much information to write to the file, and whether to append new information or overwrite old information.

Adding a NameServer with the Progress Explorer

Use the following steps to add a NameServer:

3. Start the OpenEdge Explorer..
4. Select the Resources Tab.
5. Select New OpenEdge Resource
6. Select NameServer
7. Type the name of the NameServer.
8. Select Location Local.
9. Select Save.
   You will be taken to a new page showing the available settings



Resources > OpenEdge > NameServer > NS2 > Configuration

NameServer Configuration: mhlaptop.NS2

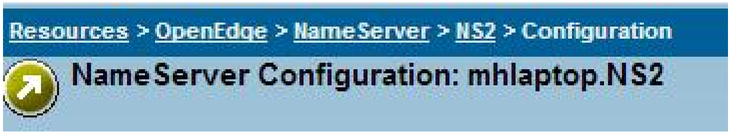
Location

Location: local
Host name: localhost
Port number: 5162

10. Click on Edit.

11. Enter a Port Number.

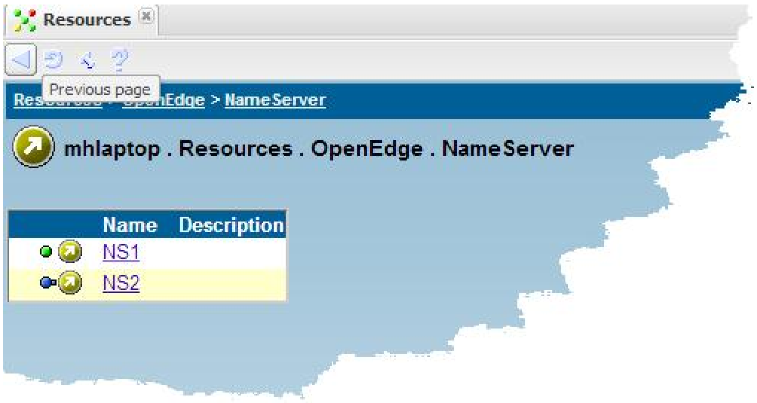
12. Choose Logging Setting.
13. Enter the server log filename.
14. Disable Append to NameServer log file.
15. Click on Save.
16. Click on 'NameServer in the Resource list



Resources > OpenEdge > NameServer > NS2 > Configuration

NameServer Configuration: mhlaptop.NS2

17. Verify that the NameServer displays in the list of NameServers



Resources > OpenEdge > NameServer

mhlaptop . Resources . OpenEdge . NameServer

Name	Description
NS1	
NS2	

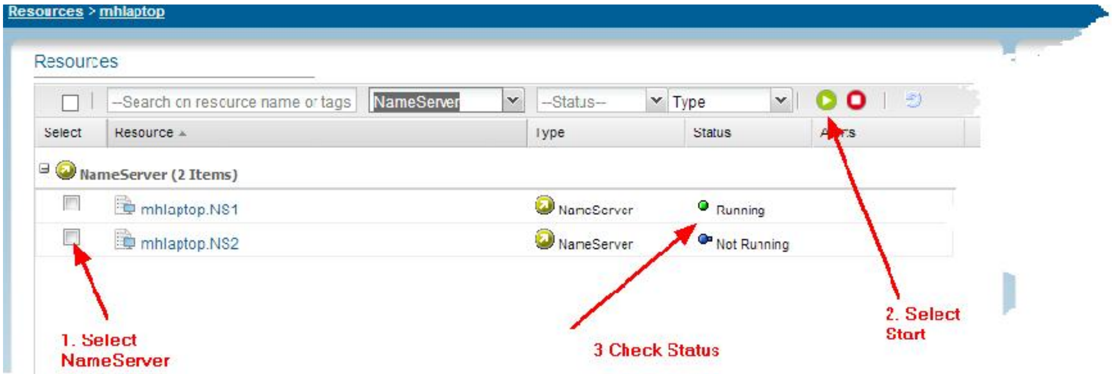
Starting a NameServer

© DataPA

Manually starting a NameServer

Follow these steps to start a NameServer on your system, At the default resources page of the OpenEdge Explorer

1. Select your NameServer in the Resource column.
2. Click on the Green Start button
   Note: Allow time for the NameServer to start.
3. Look at the Status column to see if the NameServer started Ok.



Select	Resource	Type	Status	Alerts
NameServer (2 Items)				
□	mhlaptop.NS1	NameServer	Running	
□	mhlaptop.NS2	NameServer	Not Running	

Changing to auto start

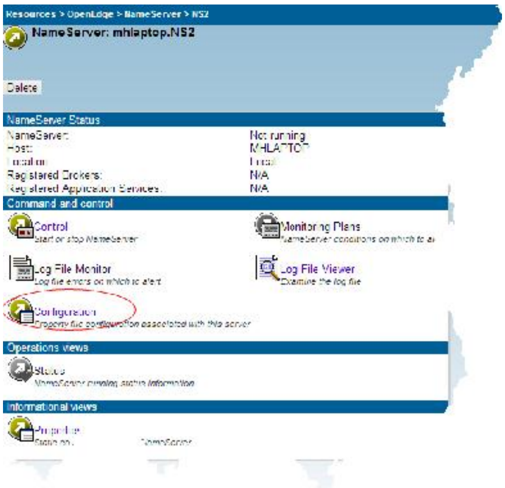
When the AdminServer starts, it looks for any other components set to autostart. When all components are set to autostart, the AdminServer starts databases first, then NamerServers, and finally AppServers. Use the following procedure to set the Nameserver to autostart the next time the AdminServer starts:

1. Follow these steps to change to auto start:

2. From the Resources pages, select the Nameserver

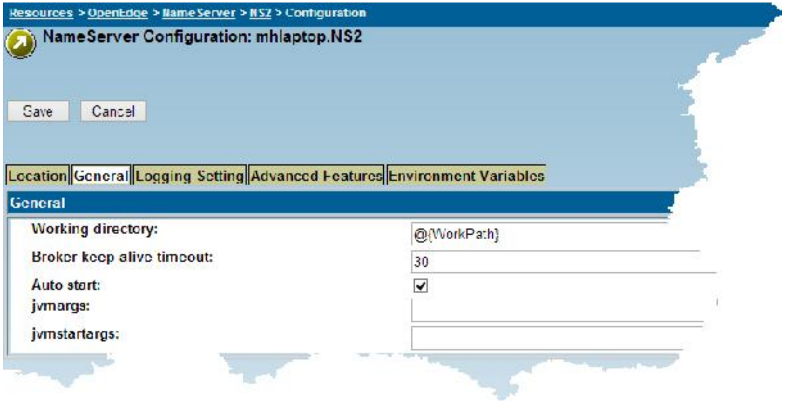
3. Select the Configuration link from the NameServer Page

4. Select Edit
5. Select the General Tab



NameServer:	Not running
Host:	MHLAPTOP
Location:	Local
Registered Brokers:	N/A
Registered Application Services:	N/A

6. Check the Auto-Start option to Enable Auto start


Resources > OpenEdge > NameServer > NS2 > Configuration
NameServer Configuration: mhlaptop.NS2
Save
Cancel
Location
General
Logging Setting
Advanced Features
Environment Variables
General

Working directory:	@{WorkPath}
Broker keep alive timeout:	30
Auto start:	☑
jvmargs:	
jvmstartargs:	

7. Choose Save.

NameServer Properties

Location

Value	Description
NameServer Location	A read-only property that indicates whether the NameServer is local or remote: Local service -- The NameServer runs locally on the selected host. You can configure all the properties of a local NameServer. Remote service -- The NameServer runs remotely on a network machine that is separate from the selected host. You can only set this property when you first create the NameServer instance. To change the location of a NameServer, you must first delete and recreate the NameServer with the new location setting.
Host name	The name of the host. If the NameServer is local, this is set read-only to 'localhost'. Otherwise, enter the remote host name where the NameServer is to run.
Port number	The number of the UDP port that the NameServer will use to listen for client connection requests and registration messages from AppServers, DataServers, and WebSpeed Transaction Servers.

General

Value	Description
Working directory	The NameServer working directory. Enter the pathname or choose the Browse button to select a pathname from the system.
Broker keep alive timeout	A value, in seconds, that indicates how often the NameServer should check for Unified Broker instances that have timed out. When a Progress Unified Broker instance registers with a NameServer, the instance indicates how often it will send "keep-alive" messages by setting a registration retry value (a property setting in Advanced Features for the Unified Broker). Once a NameServer determines that it has not received a "keep-alive" message from a Unified Broker instance within the broker's registration retry time, the NameServer automatically unregisters the instance. Note that you should use a Broker keep alive timeout value that is somewhat larger than the Unified Broker instance's registration retry value. The NameServer adjusts the specified value to allow for normal networking delays that can occur within your computer network. To keep the NameServer from using up computer resources unnecessarily, set the Broker keep alive timeout to a value that is at least 30% larger than the typical Unified Broker registration retry value.
Auto start checkbox	A checkbox that you can select if you want the NameServer to start automatically when the controlling AdminServer starts.

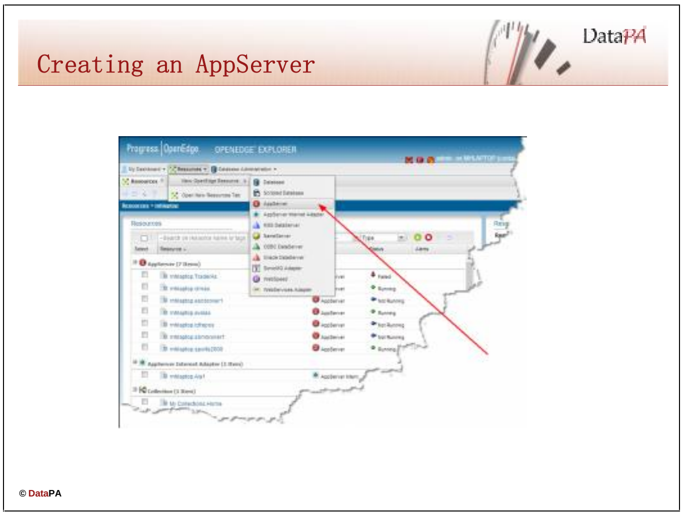
Logging Setting

Value	Description
Server log filename	The NameServer log filename. Enter a valid pathname or choose the Browse button to select the pathname from the system.
NameServer logging level	A value that specifies the amount of information to be written to the server log. Select from the following values in the drop-down list: Error only - If you are concerned about log size, choose this option. Only error-related information is written to the log file. Terse - Limited information beyond error-related information is written to the log file. Verbose - (Default) Information on all server activity is written to the log file. This option produces a large server log file whose size might require monitoring. Enable this setting for troubleshooting.
Append to the NameServer log file	A check box that indicates whether a new server log file should be created when the NameServer is started. To create a new server log file each time the NameServer is started, even if the server log file already exists, select the Append to NameServer log file check box (the Default). To append log entries to the existing server log file, clear the Append to NameServer log file check box.

NameServer Properties (Continued..)

Advanced Features

Value	Description
Neighbouring NameServers	A list of selected NameServers to which this NameServer can forward connection requests for Application Services that are not registered with it (that is, the Application Service name is unknown). The list contains all NameServer instances defined in the Progress Explorer, initially with none selected. To make a NameServer a neighbouring NameServer, select the unselected NameServer to highlight it. You can select as many NameServer instances as you want to be neighbouring NameServers. Each neighbouring NameServer thus appears highlighted in the list. To remove a NameServer from the list of neighbouring NameServers, select the already-selected NameServer to remove the highlight. The setting of this property is optional. When a NameServer receives a request for an Application Service name that is unknown and there are no neighbouring NameServers specified, the NameServer sends a message to the requesting client indicating that the Application Service is unknown. When such a NameServer has neighbouring NameServers specified, it forwards the request to each of the neighbouring NameServers. The first neighbouring NameServer that has the Application Service name, and responds to the connection request, provides the specified connection to the client.



Creating an AppServer

© DataPA

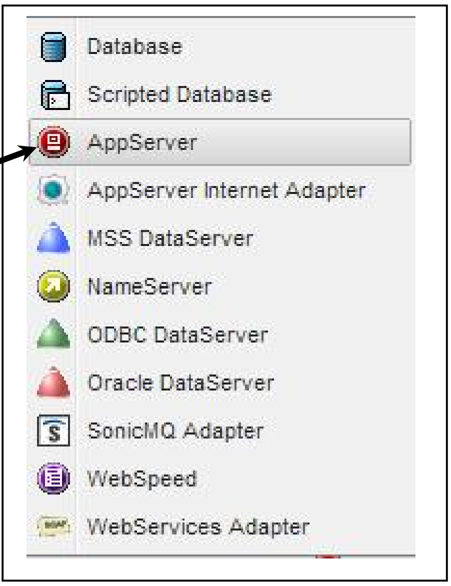
AppServer instances

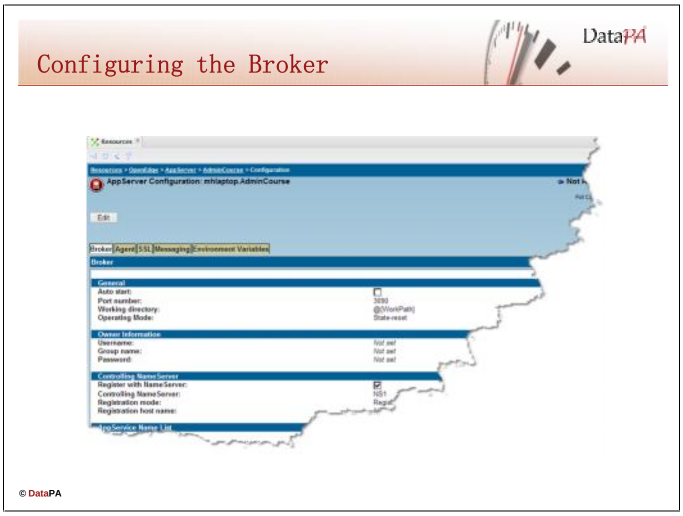
AppServer instances include an AppServer Broker and a pool of AppServer Agents. The AppServer Agents run on the AppServer host and execute ABL procedures in response to DataPA client requests. The AppServer Broker manages the DataPA connections and dispatches requests to AppServer Agents.

Creating an AppServer

To create an AppServer, follow these steps:

1. Start the OpenEdge Explorer..
2. Select the Resources Tab.
3. Select New OpenEdge Resource
4. Select AppServer
5. Type the name of the AppserverServer
6. Select Save
7. The page will refresh to the Broker Configuration Page





Configuring the Broker

AppServer Configuration: mhlaptop.AdminCourse

Broker

General

Auto start:	□
Port number:	3090
Working directory:	@{WorkPath}
Operating Mode:	State-reset

Owner Information

Username:	Not set
Group name:	Not set
Password:	Not set

Controlling NameServer

Register with NameServer:	☑
Controlling NameServer:	NS1
Registration mode:	Regist
Registration host name:	

AppService Name List

© DataPA

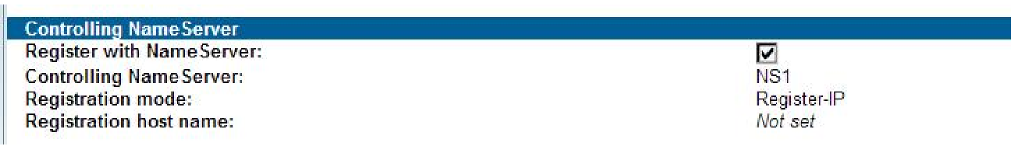
Configuring the AppServer Broker

The AppServer Properties dialog has configuration settings for the AppServer Broker and the AppServer Agents. To configure the AppServer Broker:

1. From the Resources pages, select the AppServer.
2. Select the Configuration link from the AppServer Page
3. Select Edit
4. Select the General Tab
5. Choose General to configure the following settings:

Setting	Description
Operating Mode	DataPA recommend Stateless, although all operating modes are supported.
Working Directory	Defaults to the install working directory. Can be changed to the appropriate directory on the AppServer Host.
Port Number	The number of the TCP/IP port that the AppServer Broker listens on. Entry greyed out if port already used or not allowed.
AutoStart	AppServer starts when the AdminServer starts.

6. Choose Controlling NameServer and register the AppServer with the required NameServer. Use Broker Host IP Address.



Controlling NameServer

Register with NameServer:	☑
Controlling NameServer:	NS1
Registration mode:	Register-IP
Registration host name:	Not set

7. Choose Logging Settings and select the appropriate logging settings.

AppServer Broker Properties

General

Value	Description
Operating mode	The operating mode for this AppServer. Choose the operating mode from the drop-down list. NOTE: The state-free operating mode is supported for AppServers that serve Web service clients only. **Note:** Operating modes are covered in greater detail in a later lesson.
Working directory	Your AppServer working directory. The default is the working directory set during OpenEdge installation. To change it, enter a new working directory or choose Browse to select one.
Port number	The number of the TCP/IP port that the Application Broker listens on.
Auto start	A check box indicating whether the AppServer automatically starts when the controlling AdminServer starts. Select the Auto start check box to have the AppServer start automatically.

Owner Information

Value	Description
Username	The username of an account that has system-administrative rights.
Password	The username account password.
Confirm Password	Confirmation of the username account password. Note that this field does not appear if the AdminServer is running on a non-Windows host machine.

Controlling NameServer Information

Value	Description
Register with NameServer	A checkbox that indicates whether to register the AppServer with a controlling NameServer. **Note:** Controlling NameServers are covered in more detail in the Fault Tolerance and Load-Balancing lesson.
Controlling NameServer	The name of the NameServer with which this AppServer registers. (If you did not select to register the broker with a NameServer, the field is dimmed and unavailable.)
Registration Mode	How the broker specifies its hostname if it is registering with a controlling NameServer. This hostname information is passed onto a client application when it attempts to connect to an Application Service that the broker supports. The choices are: Use Broker Host IP Address - (Default) This setting is the most efficient mechanism, and can be used in most cases. It registers with the IP address of the machine where the broker is located. Use Broker Local Host - The broker registers with the hostname of the machine that it runs on. Use this setting when the broker runs on a machine with a single hostname and more than one IP address. Use Host name - The broker registers with the values specified in the hostName property. Use this setting when your clients need a fully qualified host name to connect to a broker in a different DNS domain.

AppServer Broker Properties (Continued …)

AppService Name List

Value	Description
Application service names	A list where you can add or delete Application Service names. To add a name, enter it in the field below the list and choose Add. To delete a name, select the name in the list and choose Delete.
Supports default service	A check box that indicates whether the AppServer supports the default service. Select the Supports the default service check box to have it support the default service and clear the check box to cancel support for the default service.

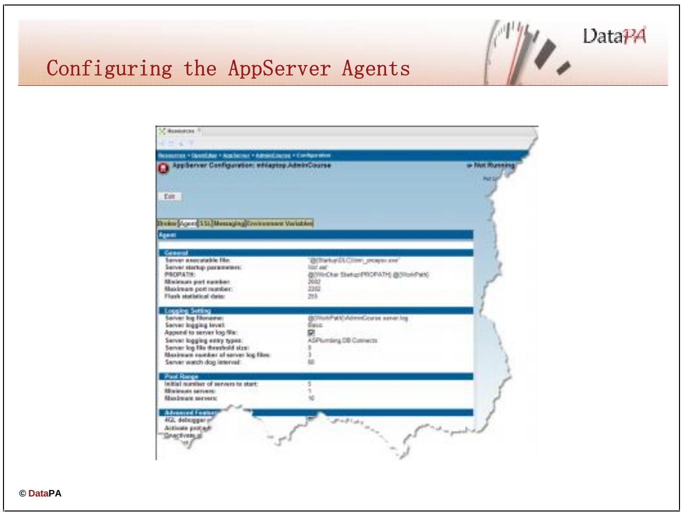
Logging Setting

Value	Description
Broker log filename	The Application Broker log filename. Enter a valid pathname or choose the Browse button to select the pathname from the system.
Broker logging level	A value that specifies the amount of information to be written to the broker log. Select from the following values in the drop-down list: Error only - If you are concerned about log size, choose this option. Only error-related information is written to the log file. Terse – Limited information beyond error-related information is written to the log file. Verbose - (Default) Information on all broker activity is written to the log file. This option produces a large broker log file whose size might require monitoring.
Append to broker log file	A check box that indicates whether a new broker log file should be created when the AppServer is started. To create a new broker log file each time the AppServer is started, even if the broker log file already exists, select the Append to broker log file check box (the Default). To append log entries to the existing broker log file, clear the Append to broker log file check box.
Logging entry types	The only supported entry is NSPlumbing. Specifying the NSPlumbing log entry type turns on logging for different NameServer actions based on logging level
Log file threshold size	A value of 0 or a value between 500,000 and 2,147,483,647, where 0 means there is no limit on the log file size other than what is imposed by the operating system, and any other value is the maximum size in bytes.
Maximum number of broker log files	The number of rolled-over log files to keep. The value can be 0 or a value between 2 and 999999, where 0 means there is no limit on the number of broker log files to keep. The specified number represents the maximum total number of log files to keep on disk at any time, including the current log file being written to. When the file becomes equal to or greater than the Broker log file threshold size, the client process renames it and creates a new log file. The file is renamed as follows, where ###### is a number starting at 000001 and increasing to 999999, after which it rolls back over to 000001: filename.######.extension

AppServer Broker Properties (Continued …)

Advanced Features

Value	Description
Maximum client instances	The maximum number of client connections that the broker can support concurrently. The default is 512, which is high enough to ensure that the number of client connections is virtually unlimited. Realistically, however, the system-level resources needed to support an unlimited number of clients might be exhausted before this limit is reached. If you see broker log entries that contain Exception Messages for "out of space" or "OutOfMemory" you may need to reduce the maximum number of client connections to a more reasonable value. Note, however, lowering this value can cause some client requests to be rejected ("Exceeded Max Clients"). Configuring and starting multiple brokers to handle higher client loads will alleviate the problem.
Priority weight	An integer value between 0 and 100 that influences the share of the workload that the selected AppServer will receive. The larger the value, the heavier the load that is distributed to the AppServer. The NameServer distributes client requests across all AppServers that have the same Application Service in proportion to the Priority weight value. The default is 0.
Registration retry	After registering with its controlling NameServer, the AppServer periodically sends "keep-alive" messages to the NameServer to let the NameServer know that the AppServer is still active. The Registration retry value is the number of seconds that pass between "keep-alive" messages. The default is 30.
Server startup timeout	Specifies the amount of time, in seconds, that the broker waits for an active server process to become available before starting a new server process. The default is 3.
Request timeout	Specifies the amount of time, in seconds, that the broker will wait for a server process to become available for processing a request. The default is 15. The broker waits for this period only if the Maximum servers setting has been reached. After the timeout has expired, the client receives a "no servers available" error message.
Auto-trim timeout	The amount of time, in seconds, that the broker waits before automatically trimming the number of running servers. The broker keeps track of the maximum number of servers that are simultaneously busy during the interval you specify. At the end of the interval, the broker attempts to trim the number of servers to match the Maximum busy level for the interval, or the Minimum Server Instances, whichever is greater. You can disable this feature by setting the timeout to zero (0). The default Server Auto-trim Interval is 1800 seconds. Progress Software recommends that you use this default setting to avoid unnecessary process management that might result from frequent starting and trimming of agents.



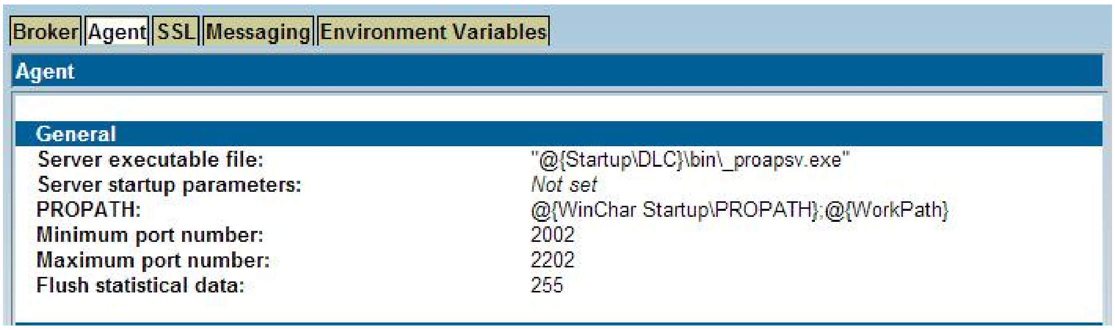
Configuring the AppServer Agents

© DataPA

Configuring the AppServer Agents

AppServer Agents, together with the AppServer Broker, make up the AppServer instance. They execute ABL procedures in response to DataPA client requests. To configure a basic AppServer Agent for DataPA, do the following:

1. Click on the Agent Tab.
2. Enter any required start-up parameters in the server start-up parameters text box to connect databases and configure each AppServer agent.
3. Add the fully qualified path to the appropriate DataPA procedure library (see table below) to the PROPATH text box.
4.

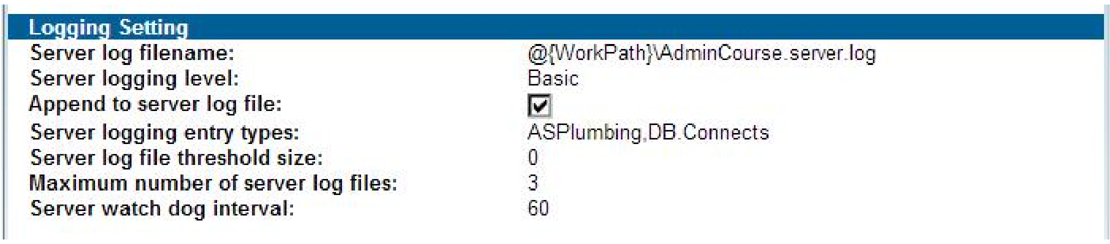


Agent

General

Server executable file:	"@{Startup\DLC}\bin_proapsv.exe"
Server startup parameters:	*Not set*
PROPATH:	@{WinChar Startup\PROPATH};@{WorkPath}
Minimum port number:	2002
Maximum port number:	2202
Flush statistical data:	255

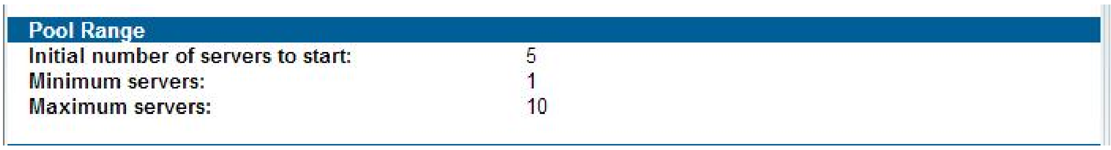
5. Enter an appropriate name and location for the log file.



Logging Setting

Server log filename:	@{WorkPath}\AdminCourse.server.log
Server logging level:	Basic
Append to server log file:	☑
Server logging entry types:	ASPlumbing,DB.Connects
Server log file threshold size:	0
Maximum number of server log files:	3
Server watch dog interval:	60

6. Choose Pool Range to define the range and quantity of Server Instances for this AppServer. For DataPA performance, you want to make sure that you set Maximum servers high enough to meet your reporting needs.



Pool Range

Initial number of servers to start: 5

Minimum servers: 1

Maximum servers: 10

Procedure Library

The table below details which procedure library you should use depending on which version of the Progress AppServer you have installed.

AppServer Version	32 Bit	64 Bit
9.1D	datapa.pl	N/A
9.1E	datapa.pl	N/A
10.0A	datapa10.pl	datapa10_64.pl
10.0B	datapa10.pl	datapa10_64.pl
10.1A	datapa10.pl	datapa10_64.pl
10.1B	datapa101B.pl	datapa101B_64.pl
10.1C	datapa101B.pl	datapa101B_64.pl
10.2A	datapa101B.pl	datapa101B_64.pl
10.2B	datapa101B.pl	datapa101B_64.pl
11.1	datapa111.pl	
11.2	datapa111.pl	
11.3	datapa111.pl	
11.4	datapa111.pl	
	datapa111.pl	

AppServer Agent Properties

General

Value	Description
Server executable file	Either the default Application Server executable pathname or a different Application Server executable file of your choosing. You generally only need to specify a non-default value if you have generated a new executable using the PROBUILD utility.
Server startup parameters	The Progress startup parameters that you want to specify to start each Application Server process.
PROPATH	The search path that AppServer Agents use to locate ABL procedures that they execute. This must include the fully qualified pathname of an appropriate DataPA procedure library (DataPA.pl for V9, DataPA10.pl for V10)
Minimum Port Number	The minimum TCP/IP port number from a range that each Application Server process can listen on. When each server process starts, it allocates a port that is not being used from the specified range up to the Maximum Port Number setting.
Maximum Port Number	The maximum TCP/IP port number port number from a range that each Application Server process can listen on. When each server process starts, it allocates a port that is not being used from the specified range down to the Minimum Port Number setting.

Logging Setting

Value	Description
Server log filename	The Application Server log filename. Enter a valid pathname or choose the Browse button to select the pathname from the system.
Server logging level	A value that specifies the amount of information to be written to the server log. Select from the following values in the drop-down list: None – No log entries. Equivalent to turning logging off. Error only - If you are concerned about log size, choose this option. Only error-related information is written to the log file. Basic- (Default) Limited information beyond error-related information is written to the log file. Verbose -Information on all server process activity is written to the log file. This option produces a large server log file whose size might require monitoring. Extended - Typically information in addition to Verbose.
Append to server log file	A check box that indicates whether a new server log file should be created when the AppServer is started. To create a new server log file each time the AppServer is started, even if the server log file already exists, select the Append to server log file check box (the Default). To append log entries to the existing server log file, clear the Append to server log file check box.

AppServer Agent Properties (Continued …)

Pool Range

Value	Description
Initial number of servers to start	The number of Application Server processes you want the Application Broker to start initially.
Minimum servers	The minimum number of Application Server processes before the Application Broker starts additional servers. If you trim the number of servers below this value, the AppServer starts any additional servers needed to maintain the specified minimum when the next client connects.
Maximum servers	The maximum number of Application Server processes that this AppServer can have running at the same time.

Advanced Features

Value	Description
ABL debugger enabled	Check box that indicates if the AppServer remote debugging facility is enabled. Select the ABL debugger enabled check box to enable the facility. If the facility is enabled, ABL client applications have the ability to step into remote AppServer procedures.
Procedures	The optional ABL procedures that you can specify to execute at strategic points during an AppServer session include the following: Activate - The name of an ABL procedure that executes before the AppServer executes a remote procedure request for a client connection that is unbound. The Activate procedure is only available for an AppServer running in stateless operating mode. Deactivate - The name of an ABL procedure that executes after the AppServer executes a remote procedure request for a client connection that is unbound. The Deactivate procedure is only available for an AppServer running in stateless operating mode. Connect - The name of an ABL procedure that executes when a client attempts to connect to this AppServer. Disconnect - The name of an ABL procedure that executes when a client disconnects from this AppServer. Startup - The name of an ABL procedure that executes when an Application Server process starts up for this AppServer. The Startup procedure is not available for an AppServer running in state-reset operating mode. Shutdown - The name of an ABL procedure that executes in an Application Server process when the Application Server process shuts down. Application Server processes shut down when you use the trim feature and when you shut down the AppServer. The Shutdown procedure is not available for an AppServer running in state-reset operating mode.
Parameters for startup procedure	The parameters to pass the Startup procedure when it executes. The Startup procedure takes a character string as an input parameter, for example: DEFINE INPUT PARAMETER startup-data AS CHARACTER NO-UNDO. You can set this parameter to any arbitrary value. If you do not specify a parameter in this field, the parameter is set to the unknown value (?) when the Application Server process executes the Startup procedure.

AppServer Agent Properties

Environment Variables

The Environment Variables category of the AppServer Property Editor allows you to add and delete environment variables. Some Environmental variables that will be described in a later lesson can be used to change the behaviour and security of DataPA. Environment variables that appear in the list are scoped to both the:

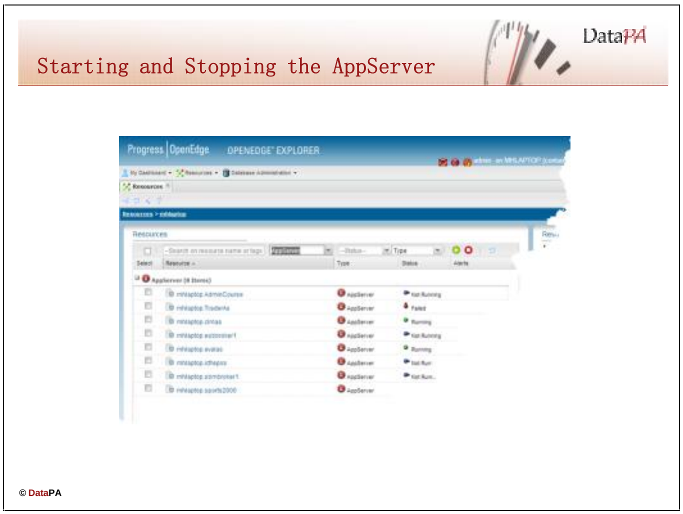
- Process in which the Application Broker executes.
- Every Application Server process that the Application Broker starts.

To add an environment variable:

1. From the property editor, choose the Environment Variable category.
2. Enter the Name and Value of the environment variable you want to set.
3. Choose Add.
4. Choose OK to save your changes.

To delete an environment variable:

1. From the property editor, choose the Environment Variable category.
2. Select the environment variable that you want to delete from the list.
3. Choose Delete.
4. Choose OK to save your changes.



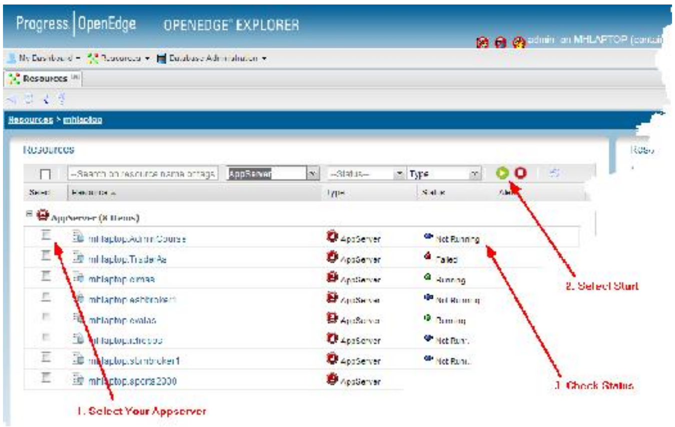
Starting and Stopping the AppServer

© DataPA

Manually Stopping the AppServer

Manual start is the default for the AppServer. Use the following steps to start an AppServer:

1. Select your NameServer in the Resource column.
2. Click on the Green Start button
   Note: Allow time for the NameServer to start.
3. Look at the Status column to see if the AppServer started Ok.



Progress | OpenEdge OPENEDGE EXPLORER
admin on MHLAPTOP (console)
My Dashboard ▾
Resources ▾
Database Administration ▾
Resources
Resources > mhlaptop
Resources
--Search on resource name or tags
AppServer
--Status--
Type

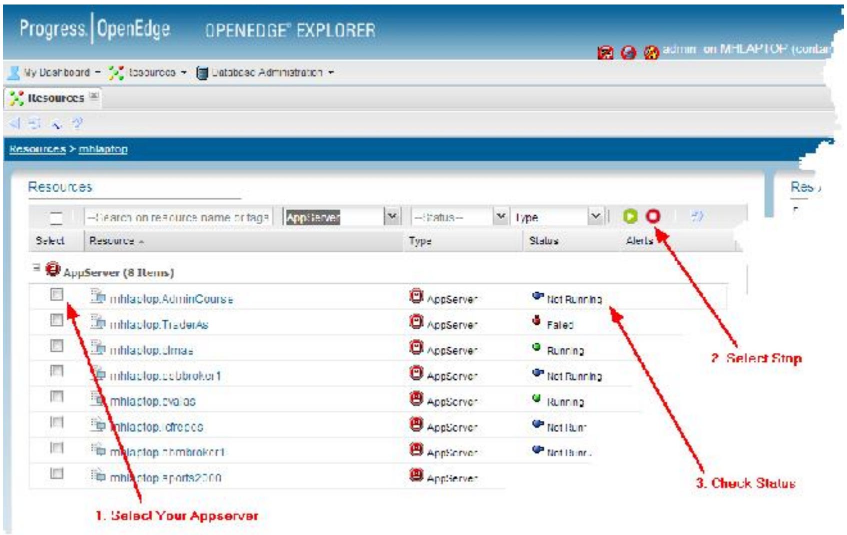
AppServer (8 items)

Select	Resource	Type	Status
□	mhlaptop.AdminCourse	AppServer	Not Running
□	mhlaptop.TraderAs	AppServer	Failed
□	mhlaptop.cimas	AppServer	Running
□	mhlaptop.ashbroker1	AppServer	Not Running
□	mhlaptop.evalas	AppServer	Running
□	mhlaptop.icfrepos	AppServer	Not Run..
□	mhlaptop.smbroker1	AppServer	Not Run..
□	mhlaptop.sports2000	AppServer	

Stopping the AppServer

Use the following steps to stop an AppServer instance:

1. Choose AppServer folder in the treeview. The list of available AppServer instances displays in the right pane.
2. Choose the Stop icon ☒ or Right-click the AppServer instance and choose Stop.



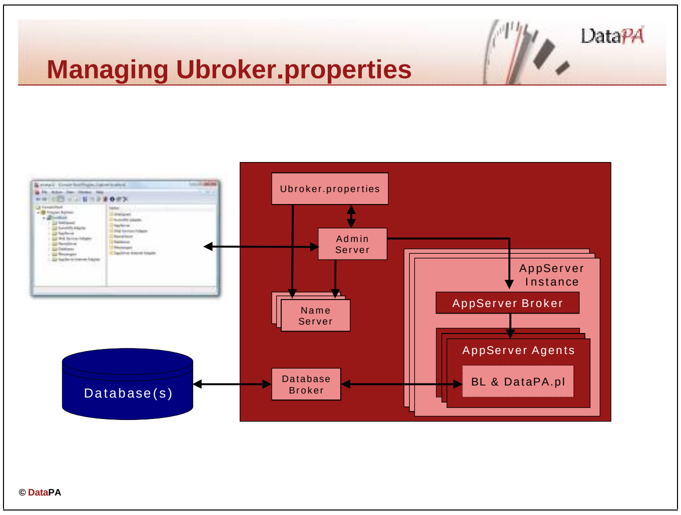
AppServer (8 Items)

Select	Resource	Type	Status	Alerts
□	mhlaptop.AdminCourse	AppServer	Not Running	
□	mhlaptop.TraderAs	AppServer	Failed	
□	mhlaptop.dmas	AppServer	Running	
□	mhlaptop.ssbbroker1	AppServer	Not Running	
□	mhlaptop.evalas	AppServer	Running	
□	mhlaptop.dfrecco	AppServer	Not Run	
□	mhlaptop.nbmbroker1	AppServer	Not Runn	
□	mhlaptop.sports2000	AppServer		

1. Select Your Appserver

2. Select Stop

3. Check Status



Managing Ubroker.properties

© DataPA

Progress Explorer writes to ubroker.properties files

When you use the Progress Explorer, it uses the AdminServer to write configuration information to the `ubroker.properties` file. As such, Progress Explorer is a graphical interface to this file.

The ubroker.properties file

The `ubroker.properties` file is a text file located in the `<%DLC%>\properties` directory on Windows platforms or `<%DLC%>/properties` on Unix and Linux platforms.

The `ubroker.properties` file stores all values that define instances of:

The NameServer

AppServer brokers

AppServer

Defaults

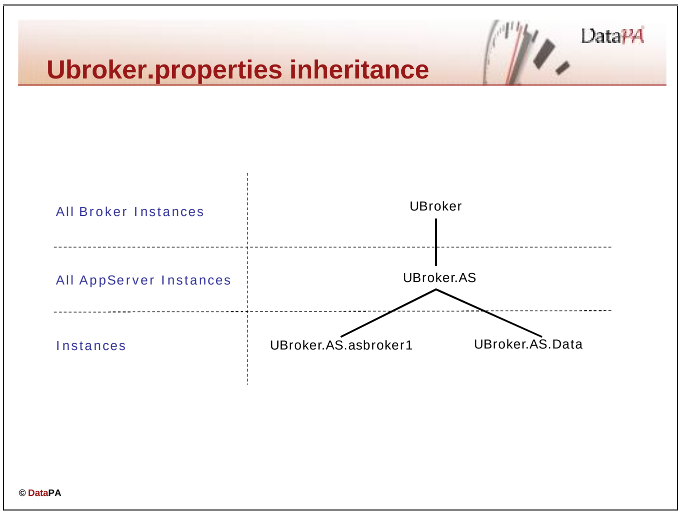
The `ubroker.properties` file contains default property values. The table below shows some of these defaults.

Property	Default value
NameServer	NS1
AppServer Broker	asbroker1

These default values display in the Progress Explorer. When the Progress Explorer starts up, it reads configuration defaults from `ubroker.properties`.

Editing ubroker.properties

While you can edit ubroker.properties in a text editor, it is recommended that you use the Progress Explorer to avoid inadvertently corrupting the file. If the ubroker.properties file is on a UNIX system, it is recommended that you update it remotely using Progress Explorer.



Ubroker.properties inheritance

All Broker Instances

All AppServer Instances

Instances

UBroker

UBroker.AS

UBroker.AS.asbroker1

UBroker.AS.Data

© DataPA

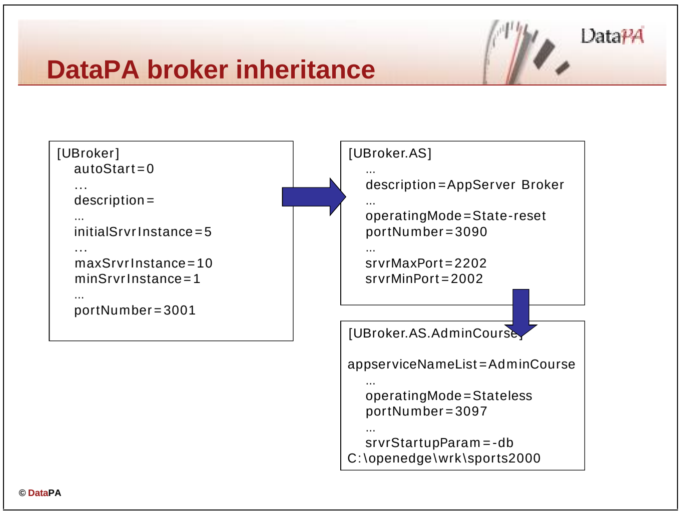
Configuration inheritance

The configuration settings are organized so that more specific instances inherit settings from less specific instances; that is, a specific AppServer broker, such as asbroker1, will inherit many of its property settings from a more generic AppServer broker setting, UBroker.AS, and the even more generic broker setting, Ubroker.

Start with basic settings for all brokers

The basic settings for all instances of all brokers (AppServer, WebSpeed and DataServer) are set under [UBroker]. Note the defaults for server instances that you saw when you created a new AppServer.

```
[UBroker]
   ABLSrcCompile=0
   appserviceNameList=
   autoStart=0
   autoTrimTimeout=1800
   brkrLogAppend=1
   brkrLogEntries=0
   brkrLogEntryTypes=UBroker.Basic
   brkrLoggingLevel=2
   brkrLogThreshold=0
   brkrNumLogFiles=3
   brokerLogFile=@{WorkPath}\broker.log
   certStorePath=@{Startup\DLC}\certs\
   collectStatsData=0
   connectingTimeout=60
   controllingNameServer=
   defaultService=0
   description=
   environment=
   flushStatsData=255
   groupName=
   hostName=
   infoVersion=9010
   initialSrvrInstance=5
   jvmArgs=
   keyAlias=
   keyAliasPasswd=
   keyStorePasswd=
   keyStorePath=@{Startup\DLC}\keys\
   maxClientInstance=512
   maxSrvrInstance=10
   minSrvrInstance=1
   …
```



DataPA broker inheritance

```
[UBroker]
   autoStart=0
   ...
   description=
   ...
   initialSrvrInstance=5
   ...
   maxSrvrInstance=10
   minSrvrInstance=1
   ...
   portNumber=3001
```

```
[UBroker.AS]
   ...
   description=AppServer Broker
   ...
   operatingMode=State-reset
   portNumber=3090
   ...
   srvrMaxPort=2202
   srvrMinPort=2002
```

```
[UBroker.AS.AdminCourse]

appserviceNameList=AdminCourse
   ...
   operatingMode=Stateless
   portNumber=3097
   ...
   srvrStartupParam=-db
C:\openedge\wrk\sports2000
```

© DataPA

Example: Add or override settings for all AppServer brokers

Any new property settings or overrides specifically for all AppServer brokers are set under [UBroker.AS].

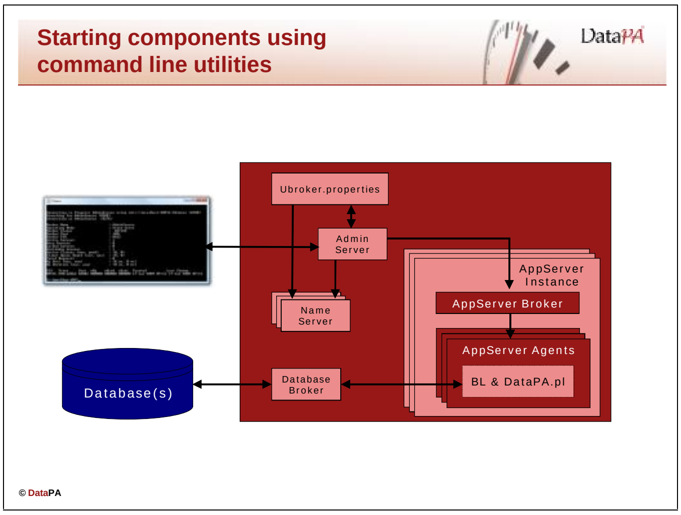
```
# [UBroker.AS]
    ABLSrcCompile=0
    …
    debuggerEnabled=0
    defaultService=0
    description=AppServer  Broker
    …
    mqSrvrLogAppend=1
    mqSrvrLogEntries=0
    mqSrvrLogEntryTypes=UBroker.Basic
    mqSrvrLoggingLevel=2
    mqSrvrLogThreshold=0
    mqSrvrNumLogFiles=3
    operatingMode=State-reset
    portNumber=3090
    …
```

Example: Add or override settings for a specific AppServer broker

New property settings or overrides for specific AppServer brokers (in this example, DataPA) are set for that broker under [UBroker.AS. DataPA].

Here is the entry in ubroker.properties that describes the DataPA AppServer broker:

```
 [UBroker.AS.AdminCourse]
    appserviceNameList=AdminCourse
    brokerLogFile=@{WorkPath}\DataPA.broker.log
    controllingNameServer=PANamesServer
    mqBrokerLogFile=@{WorkPath}\AdminCourse.mqbroker.log
    mqServerLogFile=@{WorkPath}\AdminCourse.mqserver.log
    operatingMode=Stateless
    portNumber=3097
    …
    srvrLogFile=@{WorkPath}\AdminCourse.server.log
    srvrStartupParam=-db C:\openedge\wrk\sports2000
    uuid=40f8f428dd1eaaaa:cdedfd:11469301ce5:-73d8
```



Starting components using command line utilities

When to use the utilities

As an alternative to using the Progress Explorer to start and monitor AppServer components, you can use the command line utilities. These utilities are primarily for UNIX (non-graphical) users. However, they can be used on Windows platforms from a CMD window.

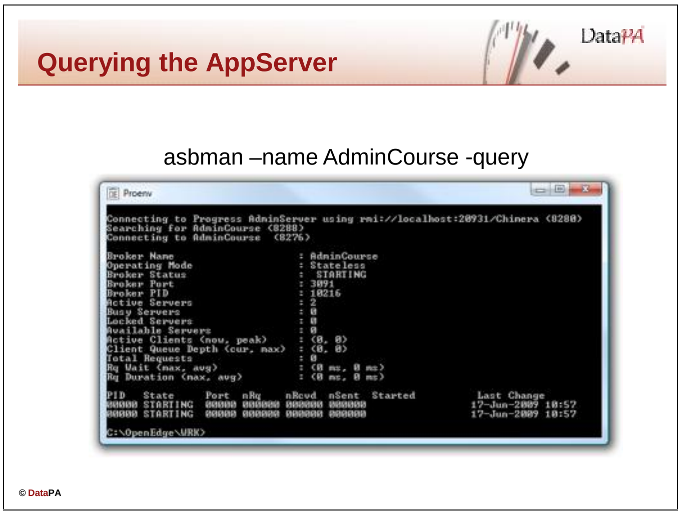
Command line utilities

The command line utilities for AppServers are:

Name	Description
dbman	Start, stop, and query a database
nsman	Start, stop and query the NameServer
nsconfig	List and validate configuration values for the NameServer
asbman	Start, stop and query an AppServer broker
asconfig	List and validate configuration values for an AppServer broker
proadsv	Start and stop the AdminServer

For more information

The next pages give an introduction to the command line utilities. For more information on these utilities see the *OpenEdge Application Server: Administration* and the *OpenEdge Getting Started: Installation and Configuration for Unix* or the *OpenEdge Getting Started: Installation and Configuration for Windows*



Querying the AppServer

asbman –name AdminCourse -query

```
Connecting to Progress AdminServer using rmi://localhost:20931/Chimera (8280)
Searching for AdminCourse (8288)
Connecting to AdminCourse  (8276)

Broker Name                        : AdminCourse
Operating Mode                     : Stateless
Broker Status                      :  STARTING
Broker Port                        : 3091
Broker PID                         : 10216
Active Servers                     : 2
Busy Servers                       : 0
Locked Servers                     : 0
Available Servers                  : 0
Active Clients (now, peak)         : (0, 0)
Client Queue Depth (cur, max)      : (0, 0)
Total Requests                     : 0
Rq Wait (max, avg)                 : (0 ms, 0 ms)
Rq Duration (max, avg)             : (0 ms, 0 ms)

PID   State     Port  nRq    nRcvd  nSent  Started           Last Change
00000 STARTING  00000 000000 000000 000000                  17-Jun-2009 10:57
00000 STARTING  00000 000000 000000 000000                  17-Jun-2009 10:57

C:\OpenEdge\WRK>
```

© DataPA

Options of the NSMAN and ASBMAN commands

The –MAN commands are used to start, stop and query the NameServer and the AppServer broker configurations. The following Job Aid shows some of the command line parameters used by the NSMAN and ASBMAN command line utilities.

NSMAN and ASBMAN parameters

Option	When to use	Definition	Required option?
-name <Instance Name>	To specify the name of the NameServer or broker	Broker or NameServer name	Yes
-start	To start the NameServer or AppServer	Starts the instance name	Yes, to start
-stop	To stop the NameServer or AppServer	Stops the instance name	Yes, to stop
-kill	To unconditionally stop the AppServer.	Emergency shutdown of AppServer.	No ASBMAN only
-user <User>	To specify the username if connecting to a remote host.	Specifies a user and password	Yes if remote host
-query	To query the process.	Queries the instance name	No
-host <Host Name>	To specify the host where the AdminServer is running.	Specifies an AdminServer host name or IP Address.	No, if local host; Yes, if remote host
-port <Port Number>	To specify the AdminServer when not using the default port.	AdminServer port number	No, defaults to 20931
-addservers <Number To Start>	To start additional AppServer agents	Number of additional agents to start	No ASBMAN only
-trimservers<Number to trim>	To stop AppServer agents.	Number of agents to stop.	No ASBMAN only

Examples

For example, to start the DataPA AppServer, use this command:

```
asbman –name AdminCourse –start
```

To stop the DataPA AppServer, use this command:

```
asbman –name AdminCourse –stop
```

To query the DataPA AppServer, use this command:

```
asbman –name AdminCourse –query
```

Starting, stopping, and querying components from the command line

To use the command line:

1. Launch the Windows command prompt:
   - Choose Start→Run
   - Type cmd.
   - Choose OK.

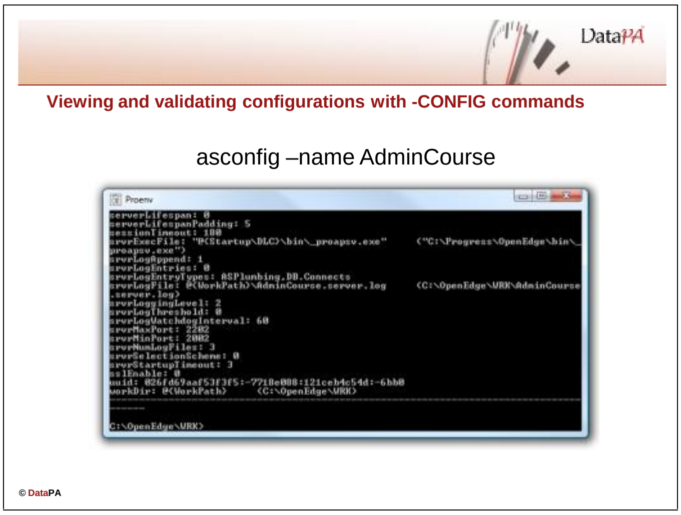
 OR

   - Choose Start→Programs→Progress→Proenv.

2. If you did not use Proenv, and if the Progress install directory is not in your Windows PATH, type:
   cd c:\Progress\OpenEdge\bin
   to change the directory to the Progress executables directory.

   You also can set your Windows PATH to include "Progress\OpenEdge\bin" either using the "set path" command or within Control Panel/System Advanced/Environment Variables. This way, you can execute a command line utility from anywhere.

3. Type asbman –name AdminCourse –query
   (commands are case-sensitive). This provides the same information as choosing Status in the Progress Explorer.



Viewing and validating configurations with -CONFIG commands

asconfig –name AdminCourse

```
serverLifespan: 0
serverLifespanPadding: 5
sessionTimeout: 180
srvrExecFile: "@{Startup\DLC}\bin\_proapsv.exe"     {"C:\Progress\OpenEdge\bin\_
proapsv.exe"}
srvrLogAppend: 1
srvrLogEntries: 0
srvrLogEntryTypes: ASPlumbing,DB.Connects
srvrLogFile: @{WorkPath}\AdminCourse.server.log     {C:\OpenEdge\WRK\AdminCourse
.server.log}
srvrLoggingLevel: 2
srvrLogThreshold: 0
srvrLogWatchdogInterval: 60
srvrMaxPort: 2202
srvrMinPort: 2002
srvrNumLogFiles: 3
srvrSelectionScheme: 0
srvrStartupTimeout: 3
sslEnable: 0
uuid: 026fd69aaf53f3f5:-7718e088:121ceb4c54d:-6bb0
workDir: @{WorkPath}     {C:\OpenEdge\WRK}
--------------------------------------------------------------------------------
------

C:\OpenEdge\WRK>
```

© DataPA

Options of the –CONFIG commands

Use the –CONFIG commands to view and debug the AppServer broker, NameServer and DataServer configurations. The following chart documents the command line parameters used by the ASCONFIG and NSCONFIG command line utilities.

Option	Shortcut	Definition	Req'd
-name <Instance Name>	-i <Instance Name>	Views an existing configuration	Yes
-propfile <Path to ubroker.properties>	-f <Path to ubroker.properties>	Specifies a property file name	No
-validate	-v	Syntax checks a ubroker.properties file	No
-help	-h	Displays command line help	No

Examples

To view an AppServer broker:

```
asconfig –name AdminCourse | more
```

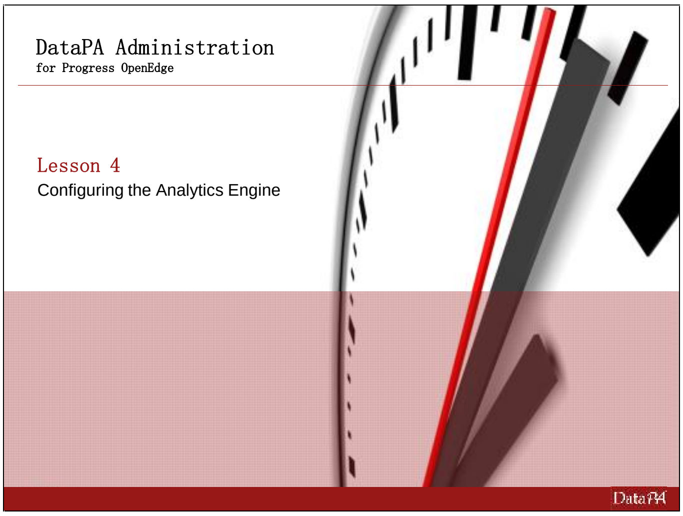
The output is piped to 'more', since there is more than one page.

Viewing and validating configurations

1. Launch the Windows command prompt.

2. Change to the Progress\OpenEdge\bin executables directory or put this directory into the Windows PATH.

3. Type asconfig –name AdminCourse | more. (The instance name is case sensitive.) The output should resemble the Ubroker.AS.AdminCourse portion of the unbroker.properties file.



DataPA Administration
for Progress OpenEdge

Lesson 4
Configuring the Analytics Engine

Lesson 4 – Configuring the Analytics Engine

Introduction

With DataPA OpenAnalytics' unique live analytics engine, IT professionals can apply their knowledge to a live data framework, allowing any user to ask innovative questions against live, operational data whilst maintaining data quality and security. Complex business logic can be expressed with the industry leading Advanced Business Language (ABL) from Progress OpenEdge. With built in data awareness, the ABL allows IT professionals to leverage business terminology and statements to quickly express rich business rules that are comprehensible by all. This lesson shows how to use the DataPA Analytics Engine to configure DataPA for use in a particular organisation. It will cover creating systems, links and subjects.

Leaning Objectives

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

- Maintain Client Data Settings
    - Create and maintain systems
    - Create and maintain links
    - Use the import and export wizards
    - Create and maintain subjects
    - Create freeform subject using the subject wizard.
    - Modify and delete existing subjects
    - Add calculated columns to a freeform subject.
    - Create, administer and use dynamic functions with a freeform subject.
    - Create Business Logic subjects from Progress Dynamic Functions
    - Create SDO subjects

Prerequisites

Before you begin this lesson you should be able to:

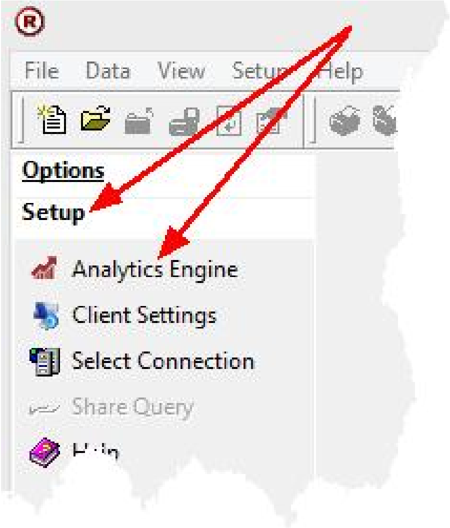
- Use DataPA Reports and DataPA in Excel and Access.
- Create a query using DataPA.
- Write ABL business logic.
- Understand the use of Dynamic functions and super procedures.

m

Introduction

The DataPA Analytics Engine screen allows you to create, configure and manage the systems, links and subjects which define the data access environment available to end users.

Opening the DataPA Analytics Engine Screen



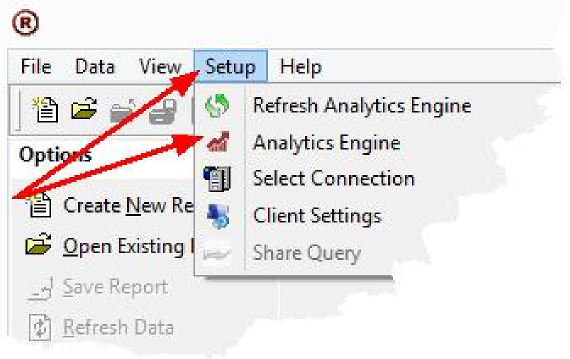
Using the Outlook Bar

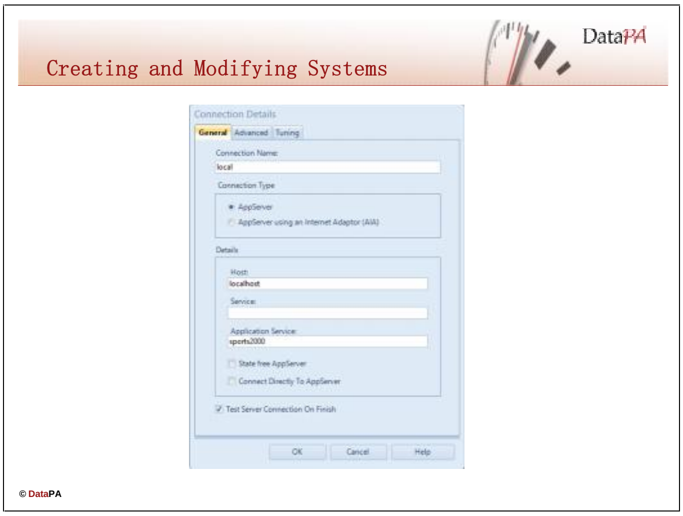
In all the DataPA interfaces you can open the DataPA Analytics Engine screen from the Outlook Bar. Click on 'Analytics Engine' to open the setup screen. If it isn't visible try first clicking on 'Setup'. If Setup is not visible you do not have Permissions to the Modify DataPA Analytics Engine (these permissions will be discussed later in this lesson).

Using the Menu bar

In all the DataPA interfaces you can open the DataPA Analytics Engine screen from the Menu Bar.

Clicking on 'Analytics Engine', available from the Setup Menu, will open the DataPA Analytics Engine Screen, shown below. If the menu option is not available you do not have permission to modify the Analytics Engine (these permissions will be discussed later in this lesson).





Creating and Modifying Systems

Connection Details

Connection Name: local

Connection Type

● AppServer

○ AppServer using an Internet Adaptor (AIA)

Details

Host: localhost

Service:

Application Service: sports2000

□ State free AppServer

□ Connect Directly To AppServer

☑ Test Server Connection On Finish

Introduction

Systems represent a single AppServer instance connection. When a user runs a report or query, DataPA connects to a specific AppServer instance, and runs Progress ABL procedures (contained in the DataPA procedure library) on the AppServer agent to retrieve the data. Systems in DataPA define the connection parameters required to connect to a specific AppServer instance and are referred to each time DataPA connects to an AppServer.

Creating a System

Follow these steps to create a new System:

1. Open the DataPA Analytics Engine screen.
2. Select *File* → '*New System*' or Click on the *'New System'* Ribbon Icon
3. Press *Next* and enter a name and description for the System.
4. Press *Next* and set *Other Details*
5. Press *Next*, then *Add* to open the *Connection Details* dialog box.
6. Enter a connection name, the name or IP address of the host machine, and the Application Service Name. Press *OK* and wait for a confirmation screen to indicate the AppServer connection was successful.
7. Press *Next*, then press *Finish*.
8. Save your Changes using the 'Save Changes' button on the Ribbon.

Modifying a System

Follow the steps below to modify an existing system:

1. Open the DataPA Analytics Engine screen and ensure your System is selected in the Tree view.
2. Select the Tab for the information you wish to change.
3. Save your Changes using the 'Save Changes' button on the Ribbon.

System Properties

Other Details

Value	Description
Allow multiple subjects in one query	Each system can be set up to allow users to combine more than one subject in a single query. The advantage is to allow the user much more flexibility, and reduce the burden on the system administrator to create an all encompassing set of subjects. For example, rather than having to create the 3 subjects Customers, Orders and Customer Orders, the administrator could just create two subjects, and allow the user to combine the customer and order subjects to produce a report with both customer and order information. The disadvantage is the assumption that users will know and correctly implement joins between two or more subjects, so in our example we would require a user to understand that customers and orders should be joined using the customer number, not the customer name which could be inaccurate and slow.
Allow users to specify lookups in query	If checked, this will allow users to add a lookup to a query in the query wizard. If not checked, lookups will only be available in a query if they have been defined against a field in the subject.

Connection Properties

General

Value	Description
Connection Name	A name by which you can recognize the connection to your system. You can set up multiple connections for a single System, for instance if you want to easily switch between a test or live version of your application, or a remote or local connection.
Test AppServer Connection on Finish	If checked, DataPA will attempt to verify the System configuration by connecting to the AppServer before allowing the wizard to continue.

Connection Types

Value	Description
AppServer	A direct connection to an AppServer, either through a NameServer or direct to the AppServer broker. Ideal where the AppServer is located on the same local area network (LAN) as the client.
AppServer using an Internet Adaptor (AIA)	A remote connection to an AppServer using http tunnelling through the Progress AppServer Internet Adaptor. Ideal where the AppServer is located on a public wide area network (WAN) such as the Internet.

Connection Properties (Continued …)

Details (AppServer)

Value	Description
Host	The name or IP address of the machine that hosts the NameServer or AppServer that the System will connect to.
Service	The port number of the process the system will connect to. By default this is the port number of the NameServer, and if the NameServer is listening to the Progress default NameServer port of 5162, this can be left blank. If you are not using the load balancing functionality of the NameServer, it is more efficient to specify that DataPA connects directly to the AppServer Broker. In this case, *Connect Directly to AppServer* should be checked, and *Service* should contain the port number that the AppServer broker is listening too.
Application Service	An Application Service Name that the required AppServer instance recognizes.
State free AppServer	If checked DataPA will connect to the Appserver state-free.
Connect Directly to AppServer	If you are not using the load balancing functionality of the NameServer, it is more efficient to specify that DataPA connects directly to the AppServer Broker. In this case, *Connect Directly to AppServer* should be checked, and *Service* should contain the port number that the AppServer broker is listening too.

Details (AppServer using an Internet Adaptor (AIA))

Value	Description
AppServer URL	Enter the string which identifies the location of the AIA (AppServer Internet Adaptor). The URL should not include usernames, passwords or http protocol identifiers. e.g. www.mydomain.com/aia/Aia
Application Service	An Application Service Name that the required AppServer instance recognizes.
Use SSL	Select if you wish DataPA to connect to the web server hosting the AIA using SSL encryption
State free AppServer	If checked DataPA will connect to the Appserver state-free.

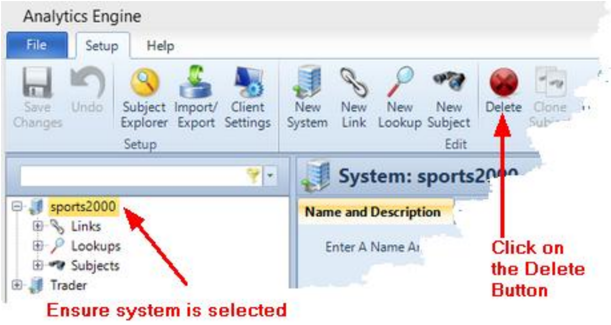
Advanced

Value	Description
AppServer Information	When DataPA connects to the AppServer, it can optionally pass *userid*, *password* and *appserver-info* arguments. These are then passed to the Connect procedure (if defined). If there is no Connect procedure the arguments are discarded. Anything entered here will be passed to the AppServer as the appserver-info argument.
Prompt user for Password	If checked, DataPA will prompt the user for a username and password each time it connects to the AppServer. This username and password will be passed to the AppServer Connect procedure.
Enable Tracing	If checked Enables Tracing of Open client Connections
Logging Level	Set the level of Logging if Enable Tracing is Enabled
Log File Name	Set the File name for the Log File is Open Client Connections are being Traced

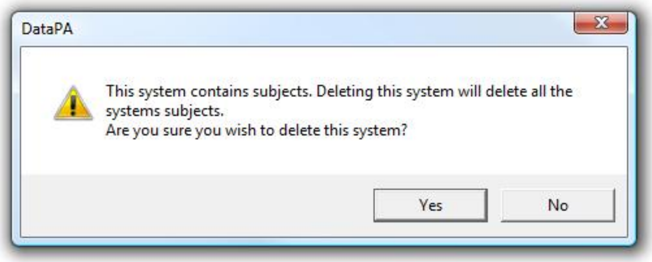
Deleting a System

Follow the steps below to delete an existing system:

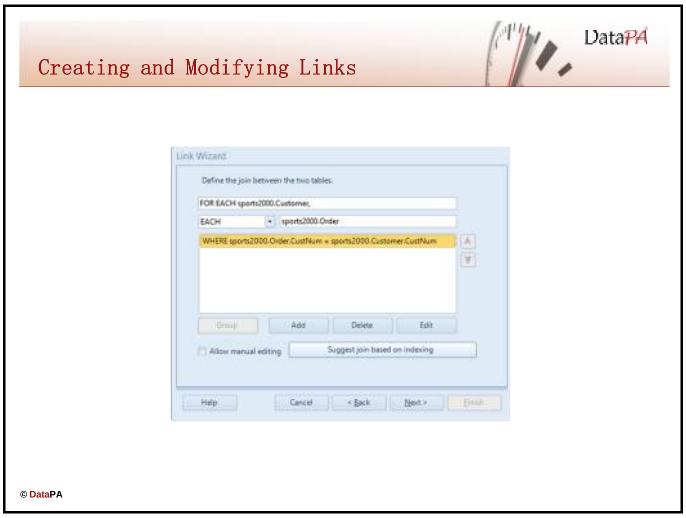
1. Open the DataPA Analytics Engine screen and ensure your System is selected in the Tree View.



2. Select *File* → *Delete* or Click on the *Delete* icon in the ribbon
3. Press *Yes*.



This system contains subjects. Deleting this system will delete all the systems subjects.
Are you sure you wish to delete this system?



Creating and Modifying Links

Link Wizard

Define the join between the two tables.

FOR EACH sports2000.Customer,

EACH sports2000.Order

WHERE sports2000.Order.CustNum = sports2000.Customer.CustNum

□ Allow manual editing

© DataPA

Introduction

A system represents a single AppServer instance that is connected to one or many databases. Links belong to a specific system, each link representing a relationship between two tables in the connected databases. When related to a Progress AppServer system, links define a Progress ABL *FOR EACH* statement that represents the relationship between two tables.

Creating a Link

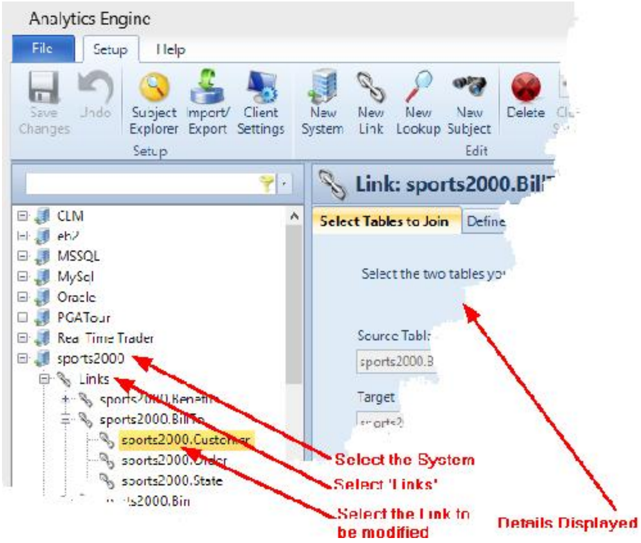
Follow these steps to create a new link:

1. Open the DataPA Analytics Engine screen the System you want to add the link to is selected in the Tree View.
2. Select *File* → *New Link* or Click on the *'New Link'* icon on the Ribbon
3. Press *Next* and select the System you wish to create a link for.
4. Press *Next*, select the two tables you wish to create a link between and select whether you wish the link to be an inner-join or an outer-join.
5. Define the join using the link condition builder. Alternatively, select *'Suggest join based on indexing'* select *Allow manual editing* and enter the required *FOR EACH* phrase.

Modifying a Link

Follow the steps below to modify an existing link:

1. Open the DataPA Analytics Engine screen and ensure your system is selected in the Tree View
2. Open the Links node
3. Select the link you wish to modify.

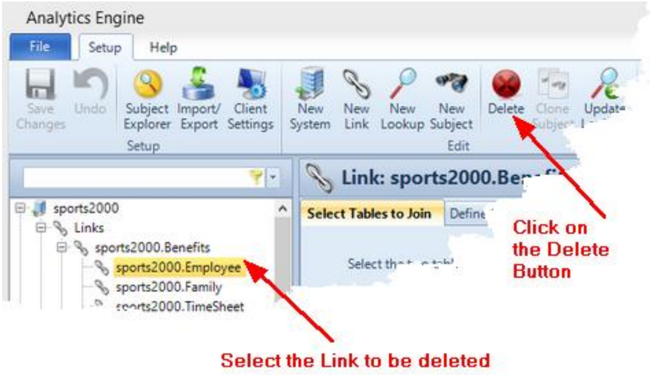


4. Select the Tab for the information you wish to change.
5. Save your Changes using the 'Save Changes' button on the Ribbon.

Deleting a Link

Follow the steps below to delete a link:

1. Open the DataPA Analytics Engine screen and ensure the System is selected.
2. Select the link you wish to delete.

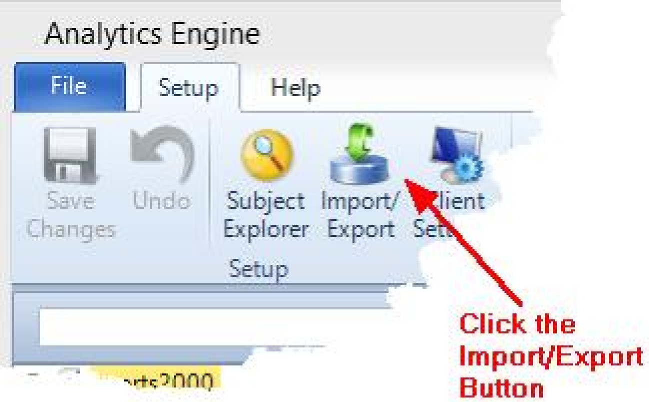


3. Select *File* → *Delete* or Click on the *Delete* icon in the ribbon

Load Default Links Wizard

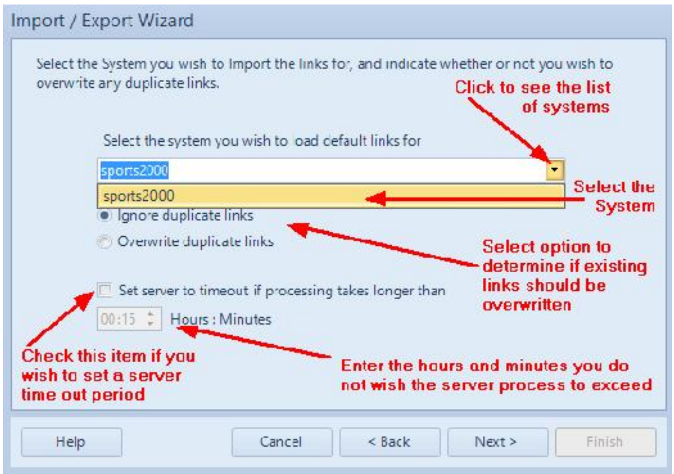
Most modern relational database applications contain many table links defined by default as part of the relational database. To import these default table relations, follow these steps:

1. Open the DataPA Analytics Engine screen
2. Select *Edit* → *Import ExportWizard*



Click the Import/Export Button

3. Press *Next* at the Introduction page
4. Select *Import Links From Database*
5. *Press Next and* select the System you wish to import links from.
6. Select whether or not you wish to overwrite duplicate links
7. Some systems contain many thousand default links. As such it may take some time to load these links, and once the server has started you cannot interrupt the process without shutting down the AppServer. To avoid the Load Default Links process taking too long you can set a timeout period. If the server process reaches the timeout period, it will terminate and no links will be loaded. To set a server timeout period, check the "Set Server timeout" checkbox, then enter the number of hours and minutes you do not wish the server process to exceed.



Select the System you wish to Import the links for, and indicate whether or not you wish to overwrite any duplicate links.

Select the system you wish to load default links for

sports2000

sports2000

● Ignore duplicate links

○ Overwrite duplicate links

□ Set server to timeout if processing takes longer than

00:15 Hours : Minutes

Click to see the list of systems

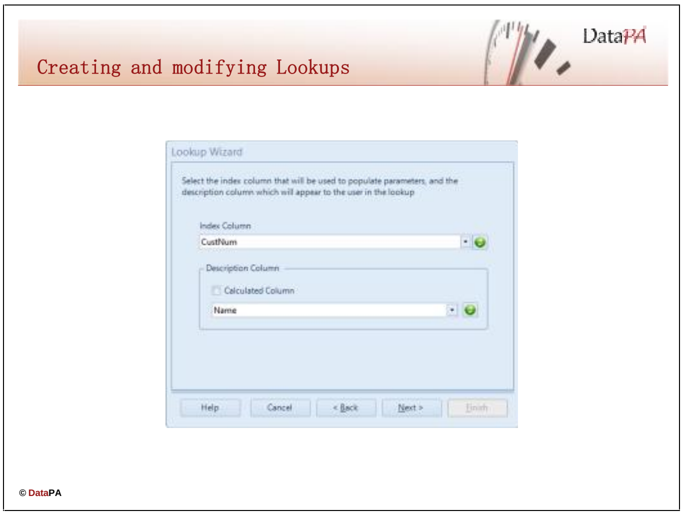
Select the System

Select option to determine if existing links should be overwritten

Check this item if you wish to set a server time out period

Enter the hours and minutes you do not wish the server process to exceed

8. Select the tables you wish to import links for.
9. Press Next to Import the Links.



Creating and modifying Lookups

Lookup Wizard

Select the index column that will be used to populate parameters, and the description column which will appear to the user in the lookup

Index Column: CustNum

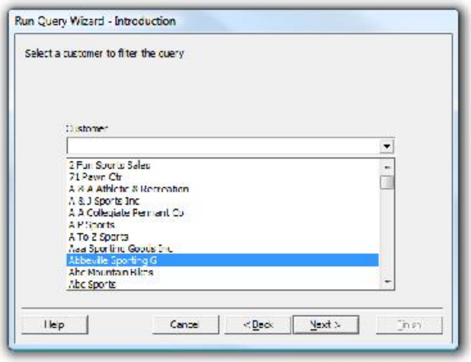
Description Column

□ Calculated Column

Name

Introduction

Lookups allow you to specify a list of values the user can select from to enter a parameter value when running a query or report. For instance, in the example below, the query requires a customer number to select a specific customer. However, the user is given a list of customer names from which to select a customer, rather than having to know and enter the customer number.



Run Query Wizard - Introduction

Select a customer to filter the query

Customer

2 Fun Sports Sales
71 Pawn Ctr
A & A Athletic & Recreation
A & J Sports Inc
A A Collegiate Pennant Co
A P Sports
A To Z Sports
Aaa Sporting Goods Inc
Abbeville Sporting G
Abc Mountain Bikes
Abc Sports

Creating a Lookup

Follow these steps to create a new lookup:

1. Open the DataPA Analytics Engine screen the System you want to add the link to is selected in the Tree View.
2. Select *File* → *New Lookup* or Click on the *New Lookup* icon on the Ribbon
3. Press *Next* at the Introduction Screen and enter a name for the lookup.
4. Press *Next* and select the System you wish to create a lookup for.
5. Press *Next* and select the table the lookup details will be created from.
6. Optionally, enter one or more filter conditions to filter the records used to populate the query. These filter conditions are similar to those that can be entered when creating a query described in lesson 2, however can only be fixed values or an expression. Press *Next.*
7. Select the column you wish to be used as the index column (the column that will contain the value used in the condition of the query).
8. Select the column you wish to use as the description (the values that the user will see in their selection), or ..
9. Select the Calculated Column check box, then press the expression button (X+1) , to open the Calculated Column dialog box. The calculated column dialog box allows you to define a calculated column using Progress ABL syntax (including calls to dynamic subjects as described in lesson 4), to build the description.
10. Press *Next,* and select the options that will apply to your lookup (see table below).
11. Press *Next* and Finish.

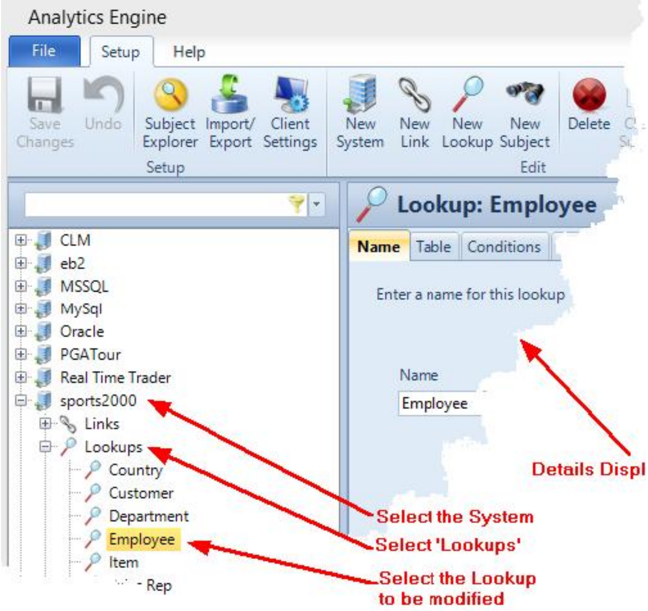
Lookup Options

Option	Description
Cache lookup data on client	Some lookups may contain a significant amount of data that can take some time to retrieve from the server. As such, DataPA will by default cache the lookup data on the client. You can force DataPA refresh this cached data by using the update lookup wizard (described below) or editing the lookup (described below). If a lookup is based on a table that will be updated frequently, you can prevent DataPA from caching the lookup by de-selecting this option.
Omit items with blank description	DataPA will by default omit any lookup entries where the description resolves to a blank string. If you want to include lookup entries with a blank string, de-select this option.
Ignore duplicate items	DataPA will by default not include multiple entries with the same description, excluding all but the first record. If you do not want DataPA to omit duplicate records, de-select this option.
Sort items	DataPA will by default sort lookup entries alphabetically based on the description. If you want lookup items to retain the order based on the index used to retrieve them from the database, de-select this item.
Allow Multiple Selections	You can allow users to select more than one item in a lookup by selecting this option. If a user select more than one item in a lookup, DataPA will build the resulting query string using an OR statement between each selected item.

Modifying a Lookup

Follow the steps below to modify an existing link:

1. Open the DataPA Analytics Engine screen and ensure your system is selected in the Tree View
2. Open the *Lookups* node
3. Select the Lookup you wish to modify.



Enter a name for this lookup

Name: Employee

4. Select the Tab for the information you wish to change.
5. Save your Changes using the 'Save Changes' button on the Ribbon.

Deleting a Lookup

Follow the steps below to delete lookup:

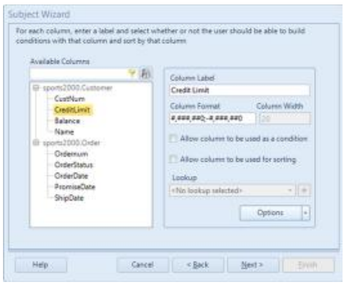
1. Open the DataPA Analytics Engine screen and ensure and ensure your system is selected in the Tree View.
2. Open the *Lookups* node
3. Select the Lookup you wish to delete.
4. Select *File* → *Delete* or Click on the *Delete* icon in the ribbon

Updating Lookups

As described above, lookups are often cached on the client to improve performance. If you know the data in a table that one or more lookups are based on has been changed, follow the steps below to force clients to refresh the lookup data;

1. Open the DataPA Analytics Engine screen
2. Select the *Update Lookups* button from the Ribbon.
3. Press *Next* to skip the introduction screen.
4. Select the system that contains the lookups you want to update and press *Next*.
5. Select the table or tables that have been updated.
6. Press *Next* and *Finish*.

Creating and Modifying Subjects



For each column, enter a label and select whether or not the user should be able to build conditions with that column and sort by that column

Column Label: Credit Limit

Column Format: #,###,##0;-#,###,##0

Column Width: 20

□ Allow column to be used as a condition

□ Allow column to be used for sorting

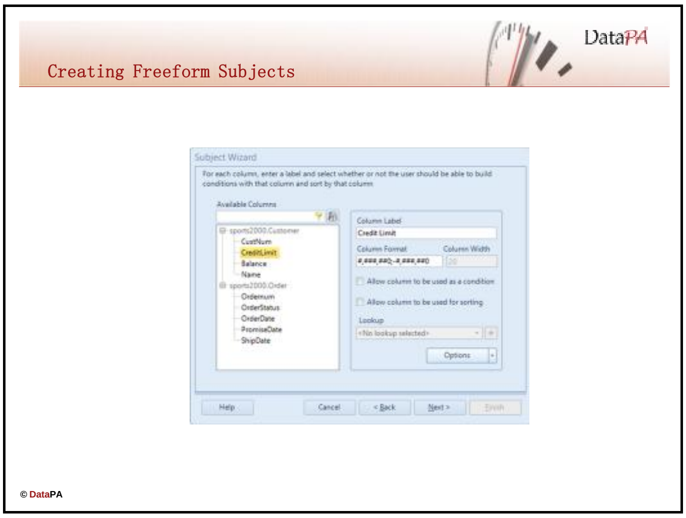
Lookup
<No lookup selected>

© DataPA

Introduction

Subjects are the foundation for all queries users will create with DataPA. As such they define the scope, structure and efficiency of the information that is available through the product.

Users create queries with DataPA to extract data for their reports and exports. Subjects are the basic building blocks for queries and all DataPA Queries are based on one or more Subjects. As such, Subjects are a way of hiding the underlying database complexity and controlling the Query building process. They provide a user friendly view of the data and prevent the use of unrelated tables, inefficient sorting on non-index fields, inefficient finds on non-index fields and the display of sensitive or system data. Each subject is related to a single system. Subjects can either be built using the links created for a system, in which case it is a freeform query, or using some bespoke ABL business logic on the server, in which case it is a Business Logic or SDO subject.



Creating Freeform Subjects

Subject Wizard

For each column, enter a label and select whether or not the user should be able to build conditions with that column and sort by that column

Column Label: Credit Limit

Column Format: #,###,##0;-#,###,##0

□ Allow column to be used as a condition

□ Allow column to be used for sorting

Lookup: <No lookup selected>

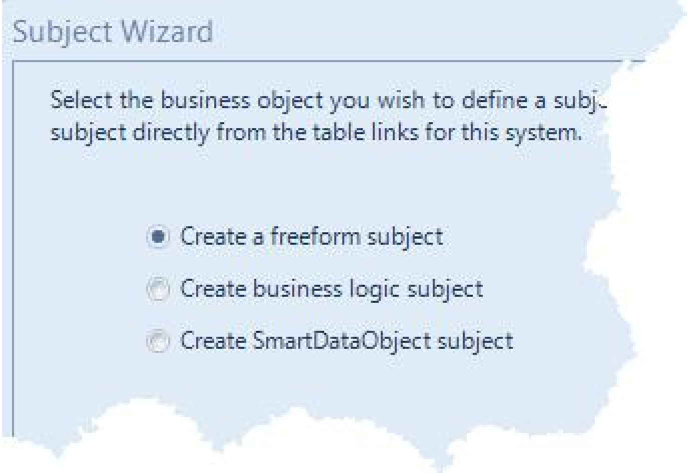
Introduction

Freeform subjects are the simplest form of subject, and should satisfy most requirements within a business application. They require little or no coding ability however still provide a flexible framework to express complex business rules. For more complex requirements, freeform subjects can be combined with server-side business logic to build complex calculations or filtering rules. Freeform subjects are built using the links created for a system.

Creating simple freeform subject

Follow these steps to create a simple freeform subject:

1. Open the DataPA Analytics Engine screen the System you want to add the Subject to is selected in the Tree View.
2. Select *File* → *New Subject* or Click on the *'New Subject'* icon on the Ribbon
3. Press *Next,* then enter a name and description for the subject.
4. Press *Next*, then select the system you wish to create a subject for
5. Press *Next,* press *Next* again to create a freeform subject



Subject Wizard

Select the business object you wish to define a subj subject directly from the table links for this system.

● Create a freeform subject

○ Create business logic subject

○ Create SmartDataObject subject

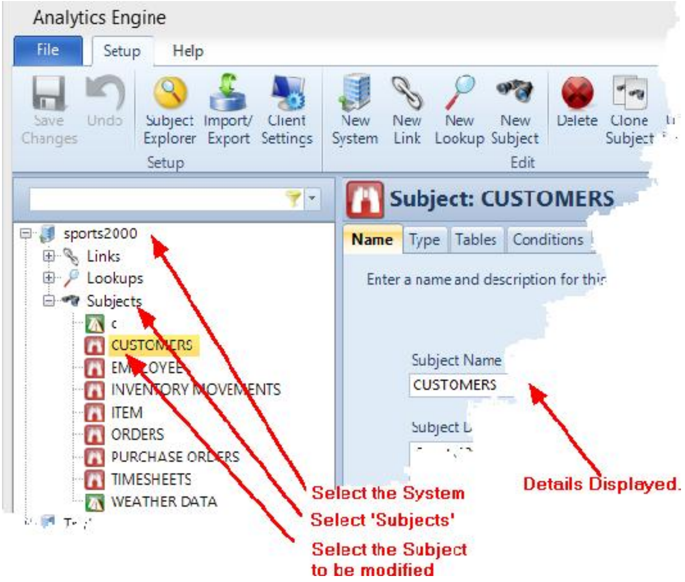
6. Select the tables you require for the subject (if you do not want to use the default link for a table, press *Modify Link* to modify the link for this subject only) and press *Next*
7. Enter any filter conditions to filter the subject. These filter conditions are similar to those that can be entered when creating a query described in lesson 2, however they are not visible to the end user and are included with any query created using this subject. Press *Next.*

8. Select the columns you wish to be available for this subject.
9. Press *Next* twice(we will cover calculated columns later in this lesson), then for each column, specify a user friendly name, whether or not it can be used as a condition and whether or not it can be used for sorting. **NB** By default, only indexed columns will be allowed as conditions and sort columns.

Modifying a Subject

Follow the steps below to modify an existing subject:

1. Open the DataPA Analytics Engine screen and ensure your system is selected in the Tree View
2. Open the Subjects node
3. Select the subject you wish to modify.



Select the System
Select 'Subjects'
Select the Subject to be modified

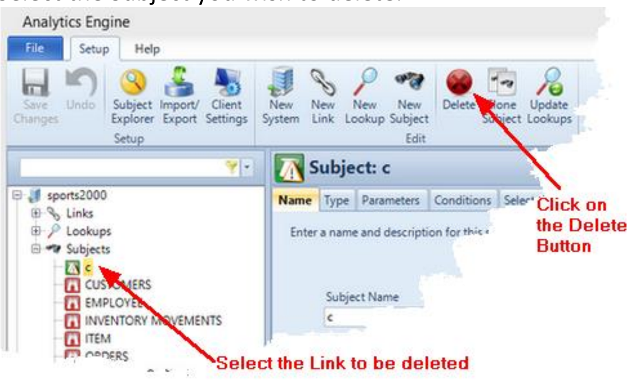
Details Displayed.

6. Select the Tab for the information you wish to change.
7. Save your Changes using the 'Save Changes' button on the Ribbon.

Deleting a Subject

Follow the steps below to delete an existing subject:

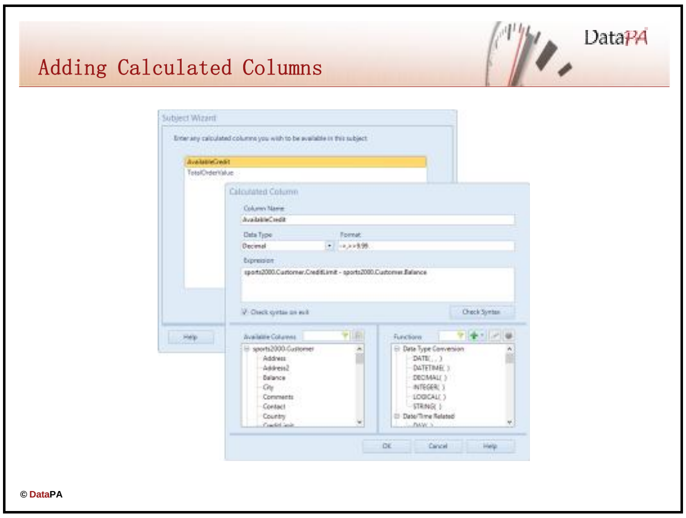
1. Open the DataPA Analytics screen and ensure the system is selected in the Tree View
2. Open the Subjects Node
3. Select the Subject you wish to delete.



Click on the Delete Button

Select the Link to be deleted

4. Select *File* → *Delete* or Click on the *Delete* icon in the ribbon



Adding Calculated Columns

© DataPA

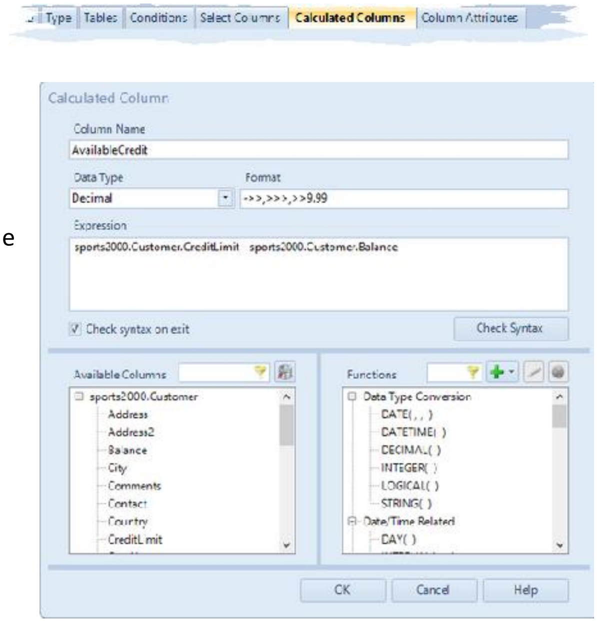
Introduction

Often you will want to add columns to a freeform subject that must be derived from a calculation rather than being directly available from the database columns. This can be done by adding calculated columns in the subject wizard.

Adding Calculated Columns to a Freeform Subject.

Follow the steps below to add a calculated field to an existing subject:

1. Open the DataPA Analytics Engine screen and select your System in the Tree view.
2. Open the Subjects node
3. Select the subject you wish to modify.
4. Select the *Calculated Columns Tab*.
5. Press *Add*
6. Enter a name for the calculated column
7. Select the appropriate Progress data type
8. Enter the Progress expression to evaluate the calculated column. The expression must be a single valid Progress expression that resolves to a single value of the data type selected. If you want to include any database columns in the expression, double-click the column in the *Available Columns* list-box.
9. Press *OK*
10. Select the *Column Attributes* Tab
11. Select each calculated field in the *Available Columns* Treeview under <*Calculated Columns*> section, then enter an appropriate label in the *Column Label* text box.
12. Press *Save Changes* on the Ribbon



Column Name: AvailableCredit

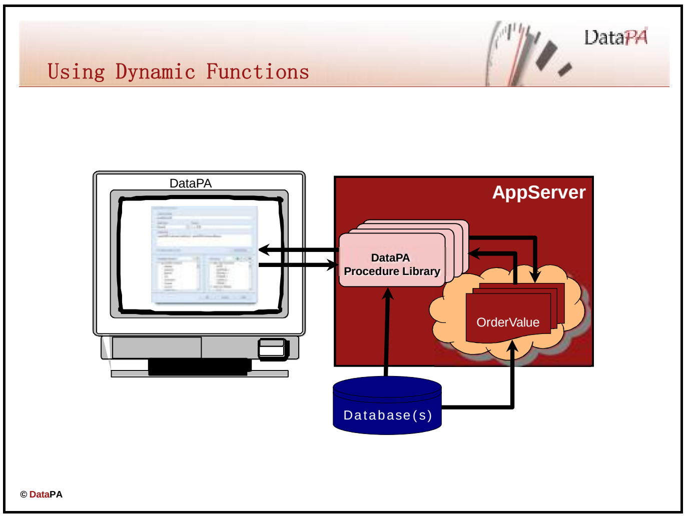
Data Type: Decimal

Format: ->>,>>>,>>9.99

Expression: sports2000.Customer.CreditLimit - sports2000.Customer.Balance

☑ Check syntax on exit

Check Syntax



Using Dynamic Functions

DataPA

DataPA

AppServer

DataPA Procedure Library

OrderValue

Database(s)

© DataPA

Introduction

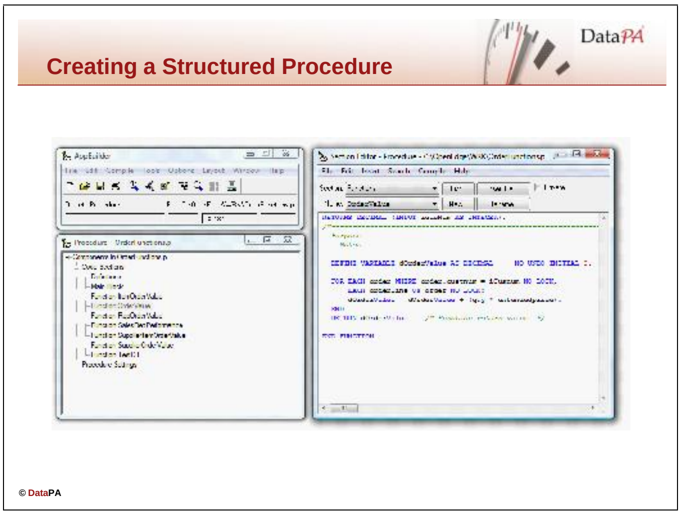
Dynamic functions can be used in DataPA to create complex calculated columns for freeform subjects and, if the function returns a handle to a temp-table, for creating Business Logic subjects.

Preparing Dynamic Functions for use by DataPA

Before you can use a dynamic function in DataPA you must complete the following steps:

1. Create a Progress structured procedure that containing the functions you wish to use from DataPA
2. Create a Progress procedure to add the structured procedure containing your functions to the super procedure stack and add this as the AppServer startup procedure.
3. Add calculated columns that call the dynamic functions to your subjects.

Each of these steps will be described in detail below.



Creating a Structured Procedure

© DataPA

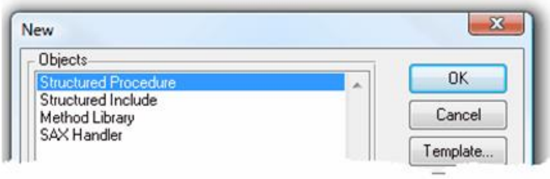
Introduction

In order to use a dynamic function in DataPA we must first write the functions as Progress ABL functions. The functions can take any number of input parameters, and return a single value.

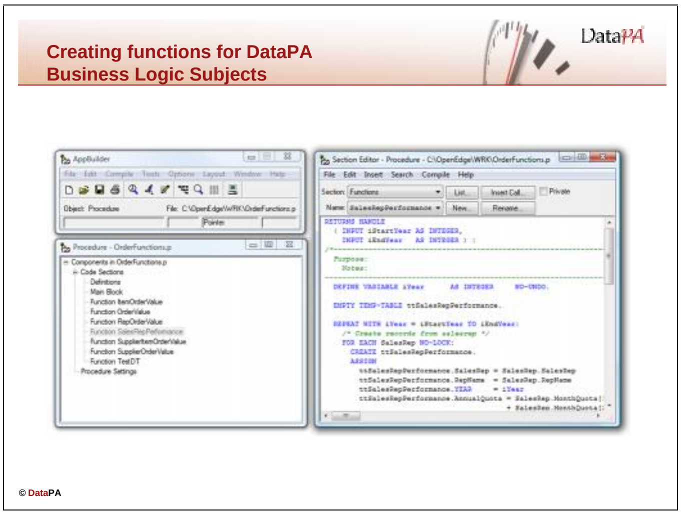
Creating a Structured Procedure

Follow these steps to create a structured procedure library that can be used for dynamic function calls from DataPA:

1. Open the Progress AppBuilder
2. Select *File*→*New*, ensure the *Procedures* tab is selected, select *Structured Procedure* and press OK
3. From the *Section* combo box of the Section Editor, select functions.
4. Press *OK* to confirm you want to create a new function.
5. Enter the name of the function you want to create and select the appropriate data type, then press *OK*.
6. Enter the code for your function, including any input parameters and setting the return value. (see example below).
7. Save the structured procedure somewhere in the PROPATH of your AppServer.



```
RETURNS DECIMAL ( INPUT iCusnum AS INTEGER ) :
/*------------------------------------------------------------------
  Purpose:
    Notes:
------------------------------------------------------------------*/
  DEFINE VARIABLE dOrderValue AS DECIMAL     NO-UNDO INITIAL 0.

  FOR EACH Order WHERE Order.custnum = iCusnum NO-LOCK:
    FOR EACH orderline OF order NO-LOCK:
      dOrderValue = dOrderValue + extendedprice.
    END.
  END.
  RETURN dOrderValue.   /* Function return value. */

END FUNCTION.
```



Creating functions for DataPA Business Logic Subjects

DataPA
AppBuilder
File Edit Compile Tools Options Layout Window Help
Object: Procedure
File: C:\OpenEdge\WRK\OrderFunctions.p
Pointer
Procedure - OrderFunctions.p
Components in OrderFunctions.p
Code Sections
Definitions
Main Block
Function ItemOrderValue
Function OrderValue
Function RepOrderValue
Function SalesRepPerformance
Function SupplierItemOrderValue
Function SupplierOrderValue
Function TestDT
Procedure Settings
Section Editor - Procedure - C:\OpenEdge\WRK\OrderFunctions.p
File Edit Insert Search Compile Help
Section: Functions
List...
Insert Call...
□ Private
Name: SalesRepPerformance
New...
Rename...

```
RETURNS HANDLE
  ( INPUT iStartYear AS INTEGER,
    INPUT iEndYear   AS INTEGER ) :
/*------------------------------------------------------------------------
  Purpose:
    Notes:
------------------------------------------------------------------------
  DEFINE VARIABLE iYear      AS INTEGER     NO-UNDO.

  EMPTY TEMP-TABLE ttSalesRepPerformance.

  REPEAT WITH iYear = iStartYear TO iEndYear:
    /* Create records from salesrep */
    FOR EACH SalesRep NO-LOCK:
      CREATE ttSalesRepPerformance.
      ASSIGN
        ttSalesRepPerformance.SalesRep = SalesRep.SalesRep
        ttSalesRepPerformance.RepName  = SalesRep.RepName
        ttSalesRepPerformance.YEAR     = iYear
        ttSalesRepPerformance.AnnualQuota = SalesRep.MonthQuota[
                                          + SalesRep.MonthQuota[
```

© DataPA

Introduction

Dynamic functions can be used in DataPA for creating Business Logic subjects. This section details how to write a Dynamic function that can be used in a DataPA Business Logic Subject.

Creating Dynamic Functions for Business Logic Subjects

To add a Dynamic Function that can be used from a DataPA Business Logic Subject, follow these steps:

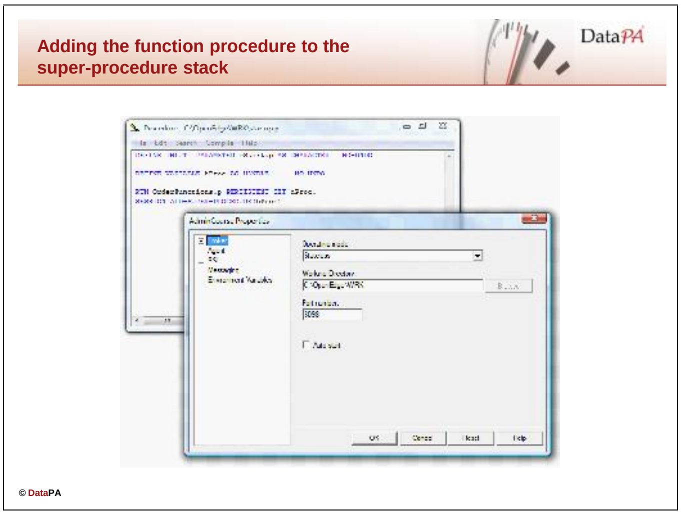
1. In the definitions section of your Structured Procedure, add a definition for the Temp-Table you wish to return. Make sure you add the RCODE-INFORMATION option to your temp-table definition so the format and labels are passed to DataPA.

```
DEFINE TEMP-TABLE ttSalesRepPerformance NO-UNDO RCODE-INFORMATION
  FIELD SalesRep    LIKE SalesRep.SalesRep
  FIELD RepName     LIKE SalesRep.RepName
  FIELD YEAR        AS INTEGER FORMAT "9999" LABEL "Financial Year"
  FIELD AnnualQuota AS DECIMAL FORMAT ">>>,>>>,>>9.99" LABEL "Annual Quota"
  FIELD TotalSales  AS DECIMAL FORMAT ">>>,>>>,>>9.99" LABEL "Total Sales"
  FIELD Percentage  AS DECIMAL FORMAT ">,>>>,>>9.99"   LABEL "Percent Achieved"
  INDEX idx1 IS PRIMARY IS UNIQUE SalesRep YEAR.
```

2. Create a new function, with a return type of handle.
3. Define any input parameters for your temp table.
4. Enter code to empty the temp-table.
5. Enter code to populate the temp-table.
6. Return the handle for the temp-table.

```
RETURNS HANDLE
  ( INPUT iStartYear AS INTEGER,
    INPUT iEndYear   AS INTEGER ) :
/*------------------------------------------------------------------------------
  Purpose:
    Notes:
------------------------------------------------------------------------------*/
  DEFINE VARIABLE iYear      AS INTEGER     NO-UNDO.

  EMPTY TEMP-TABLE ttSalesRepPerformance.

  REPEAT WITH iYear = iStartYear TO iEndYear:
    /* Create records from salesrep */
    FOR EACH SalesRep NO-LOCK:
      CREATE ttSalesRepPerformance.
```

```
          ASSIGN
            ttSalesRepPerformance.SalesRep = SalesRep.SalesRep
            ttSalesRepPerformance.RepName  = SalesRep.RepName
            ttSalesRepPerformance.YEAR     = iYear
            ttSalesRepPerformance.AnnualQuota = SalesRep.MonthQuota[1]
                                              + SalesRep.MonthQuota[2]
                                              + SalesRep.MonthQuota[3]
                                              + SalesRep.MonthQuota[4]
                                              + SalesRep.MonthQuota[5]
                                              + SalesRep.MonthQuota[6]
                                              + SalesRep.MonthQuota[7]
                                              + SalesRep.MonthQuota[8]
                                              + SalesRep.MonthQuota[9]
                                              + SalesRep.MonthQuota[10]
                                              + SalesRep.MonthQuota[11]
                                              + SalesRep.MonthQuota[12].
        END.

        /* Add up order value */
        FOR EACH Order WHERE Order.OrderDate >= DATE(1,1,iYear)
                         AND Order.OrderDate <= DATE(12,31,iYear)
                         NO-LOCK:
          FIND ttSalesRepPerformance
            WHERE ttSalesRepPerformance.SalesRep = Order.SalesRep
              AND ttSalesRepPerformance.YEAR     = iYear NO-ERROR.
          IF AVAILABLE ttSalesRepPerformance THEN DO:
            FOR EACH OrderLine OF Order NO-LOCK:
               ttSalesRepPerformance.TotalSales = ttSalesRepPerformance.TotalSales +
OrderLine.ExtendedPrice.
            END.
          END.
        END.

        /* Work out percentage */
        FOR EACH ttSalesRepPerformance:
          IF ttSalesRepPerformance.TotalSales = 0
          THEN ttSalesRepPerformance.Percentage = 0.
          ELSE ttSalesRepPerformance.Percentage = (ttSalesRepPerformance.TotalSales
/ ttSalesRepPerformance.AnnualQuota).
        END.
      END.


      FOR EACH ttSalesRepPerformance:
        DISP ttSalesRepPerformance WITH 1 COLUMN.
      END.

      RETURN TEMP-TABLE ttSalesRepPerformance:HANDLE.   /* Function return value. */

    END FUNCTION.
```



Adding the function procedure to the super-procedure stack

© DataPA

Introduction

In order to use the functions defined in structured procedure from DataPA, they need to be placed into memory on the server and made available to the AppServer. To do this, we use an AppServer startup procedure, which is called each time an agent starts, to run the structured procedure persistently and place it on the super-procedure stack of the AppServer agent session.

Adding the function procedure to the super procedure stack

Follow these steps to add the structured procedure that contains the function definitions to the super procedure stack:

1. Open the Progress AppBuilder
2. Select *Tools* → *New Procedure Window*
3. When an AppServer agent runs a startup procedure, it passes in a character variable. Therefore your procedure must start with an input character variable definition as below:

```
DEFINE INPUT PARAMETER startup-data AS CHARACTER NO-UNDO.
```

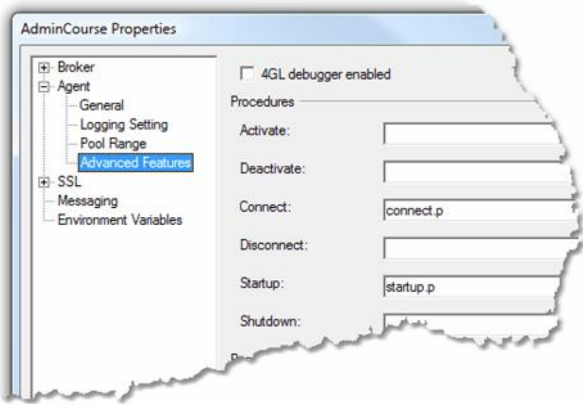
4. Define a progress handle variable we can use to store the procedure handle:

```
DEFINE VARIABLE hProc AS HANDLE      NO-UNDO.
```

6. Run the structured procedure persistently, storing the handle in the defined variable, and add it to the session super procedure stack.

```
RUN OrderFunctions.p PERSISTENT SET hProc.
SESSION:ADD-SUPER-PROCEDURE(hProc).
```

7. Save the procedure as startup.p somewhere in the PROPATH of the AppServer.
8. Open Progress Explorer
9. Open the properties screen for your AppServer
10. Select *Agent* → *Advanced Features*
11. Enter startup.p in the *Startup* text box.
12. Press *OK*
13. Restart the AppServer.



AdminCourse Properties

☐ 4GL debugger enabled

Procedures

Activate:

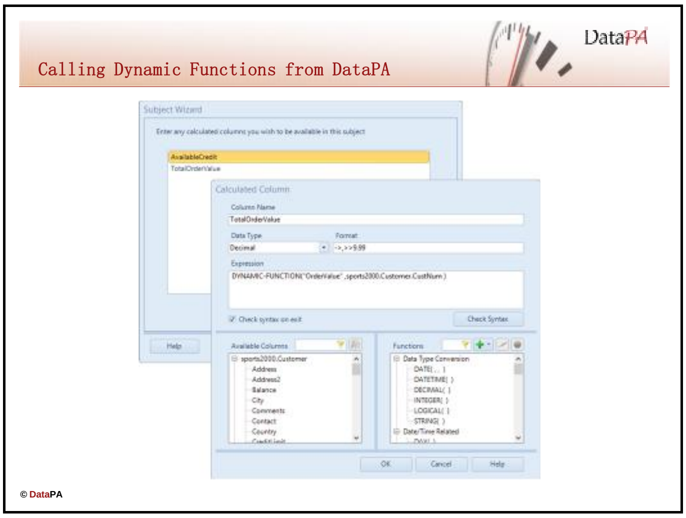
Deactivate:

Connect: connect.p

Disconnect:

Startup: startup.p

Shutdown:



Calling Dynamic Functions from DataPA

Column Name: TotalOrderValue

Data Type: Decimal

Format: ->,>>9.99

Expression: DYNAMIC-FUNCTION("OrderValue",sports2000.Customer.CustNum)

Introduction

Once a function procedure is available on the AppServer, it can be used in a DataPA subject to create calculated columns.

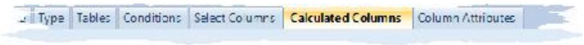
Using Dynamic Functions in Calculated Columns

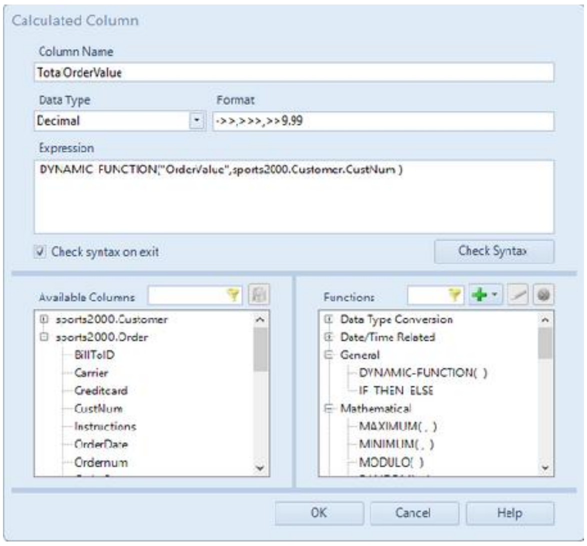
Follow the steps below to add a calculated field to an existing subject:

1. Open the DataPA Analytics Engine screen and ensure your System is selected in the Tree View
2. Select the Subjects Node.
3. Select the subject you wish to modify.
4. Select the *Calculated Columns* Tab
5. Press *Add*
6. Enter a name for the calculated column.
7. Select the Progress data type that was used when the function was defined.
8. Enter the Dynamic Function call. The call must be in the following syntax:

   DYNAMIC-FUNCTION **(** *function* **[** , *parameters* **])**

9. Press *OK*
10. Select the *Column Attributes* Tab
11. Select each calculated column in the *Available Columns* Tree view under <*Calculated Columns*> section, then enter an appropriate label in the *Column Label* text box.
12. Press *Save Changes* on the Ribbon





Calculated Column

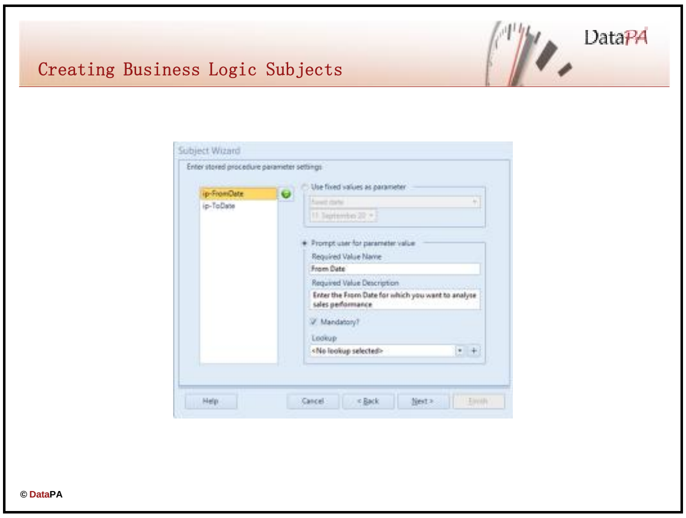
Column Name: Tota OrderValue

Data Type: Decimal

Format: ->>,>>>,>>9.99

Expression: DYNAMIC FUNCTION("OrderValue",sports2000.Customer.CustNum)

☑ Check syntax on exit



Creating Business Logic Subjects

Subject Wizard

Enter stored procedure parameter settings

ip-FromDate
ip-ToDate

○ Use fixed values as parameter

Fixed date

11 September 20

● Prompt user for parameter value

Required Value Name

From Date

Required Value Description

Enter the From Date for which you want to analyse sales performance

☑ Mandatory?

Lookup

<No lookup selected>

© DataPA

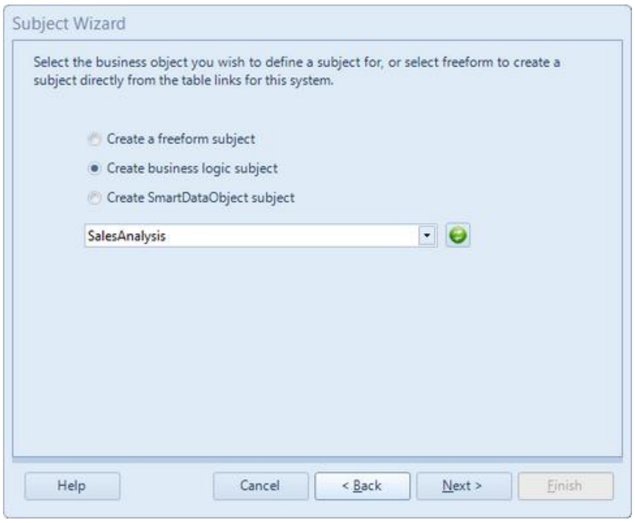
Introduction

In addition to creating freeform subjects, DataPA also allows the creation of subjects from Business Logic functions running persistently on the AppServer.

Creating Business Logic Subjects

Follow these steps to create an SDO subject:

1. Open the DataPA Analytics Engine screen and ensure your System is selected.
2. Select *File* → *New Subject* or click on the *New Subject* button on the Ribbon
3. Press *Next*, then enter a name and description for the subject.
4. Select the System and press *Next*
5. Select *Create Business Logic Subject* and select the persistently running business logic procedure you wish to use.



Subject Wizard

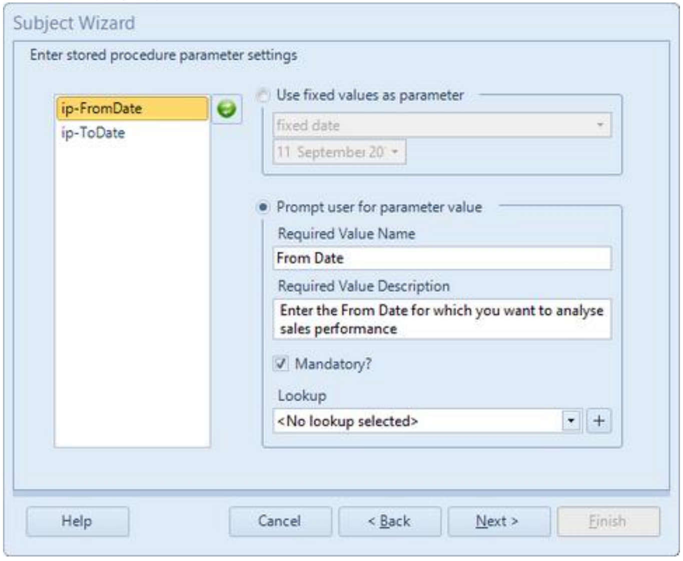
Select the business object you wish to define a subject for, or select freeform to create a subject directly from the table links for this system.

○ Create a freeform subject
● Create business logic subject
○ Create SmartDataObject subject

SalesAnalysis

6. Press *Next*.
7. For each input parameter of the business logic function, specify whether or not you want the subject to use the stored value or prompt the user for an input parameter. If you are going to prompt the user, you need to enter a name and description for the parameter and specify whether or not the parameter is mandatory. If the parameter is not mandatory, and the user chooses to skip the parameter when they run a query, a null value will be passed to the

business logic procedure.



Subject Wizard

Enter stored procedure parameter settings

○ Use fixed values as parameter

● Prompt user for parameter value

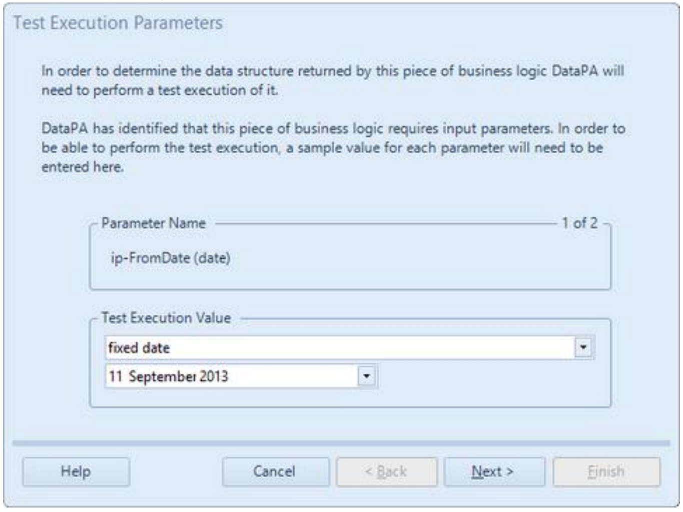
Required Value Name: From Date

Required Value Description: Enter the From Date for which you want to analyse sales performance

☑ Mandatory?

Lookup: <No lookup selected>

8. In the *Test Execution Parameter Wizard Step*, enter a value for each input parameter for your function, then press *Finish.*



Test Execution Parameters

In order to determine the data structure returned by this piece of business logic DataPA will need to perform a test execution of it.

DataPA has identified that this piece of business logic requires input parameters. In order to be able to perform the test execution, a sample value for each parameter will need to be entered here.

Parameter Name — 1 of 2

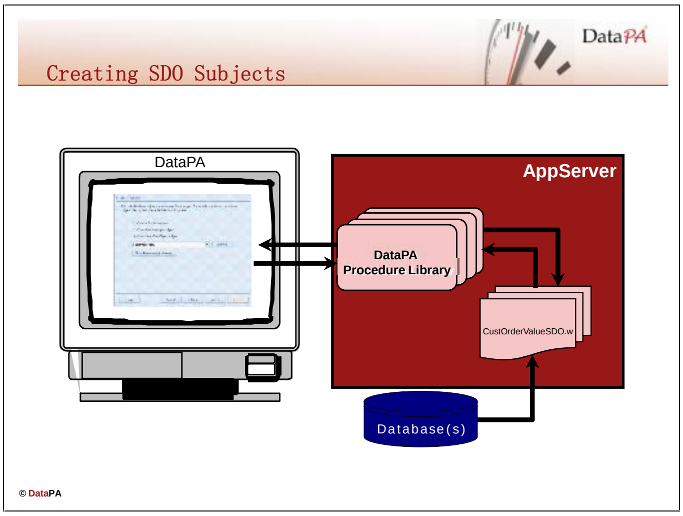
ip-FromDate (date)

Test Execution Value

fixed date

11 September 2013

9. Press *Next,* and enter any further conditions you wish to set against the subject.
10. Press *Next* and select which columns you wish to be available with the subject.
11. Press *Next*, then for each column, specify a user friendly name, whether or not it can be used as a condition and whether or not it can be used for sorting.



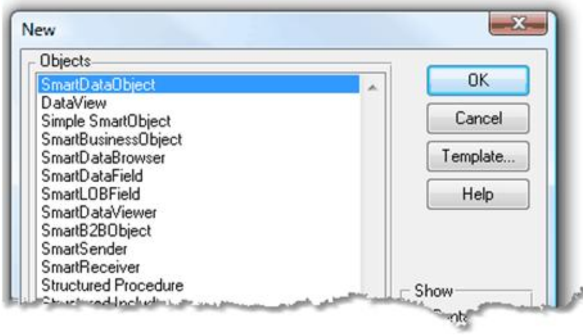
Introduction

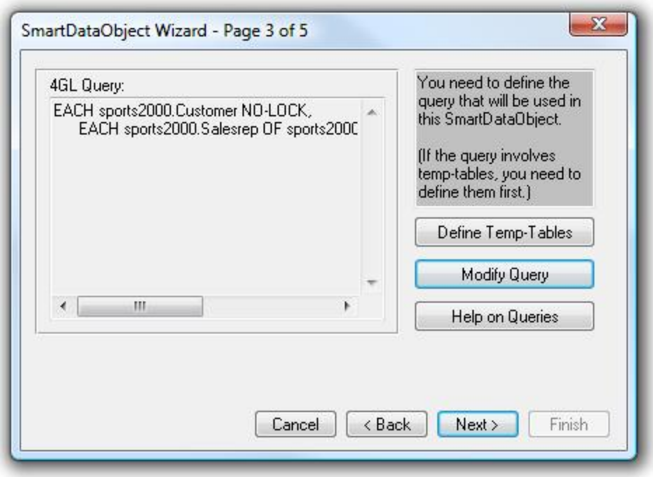
In addition to creating freeform subjects, DataPA also allows the creation of subjects from ADM2 and Dynamics Smart Data Objects (SDO's). SDO's form the basis of the Progress ADM2 and Dynamics frameworks and are designed to provide read and write access to a database in a distributed environment. For DataPA, SDO's provide a consistent method of accessing any server-side business logic to read data, allowing DataPA to support almost any complex reporting requirement. SDO's are Progress procedures and can be created quickly and simply using the Progress wizard.

Creating SDO Subjects

Follow these steps to create an SDO subject:

1. Open the Progress AppBuilder and connect to any databases required.
2. Select *File→New*
3. Make sure *SmartObjects* is checked, select *SmartDataObject* and press *OK*
4. Press *Next*
5. Press *Next* to get to the *ABL Query* screen
6. Press *Define Query*
7. Use the *Query Builder* to build your query.




SmartDataObject Wizard - Page 3 of 5

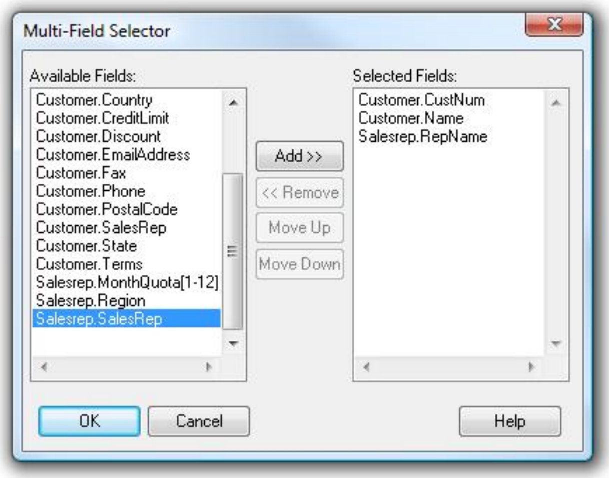
4GL Query:

```
EACH sports2000.Customer NO-LOCK,
    EACH sports2000.Salesrep OF sports200C
```

You need to define the query that will be used in this SmartDataObject.

(If the query involves temp-tables, you need to define them first.)

8. Press *Next*
9. Press *Add Fields*
10. Select the fields you wish to be available in the query and press *OK*.

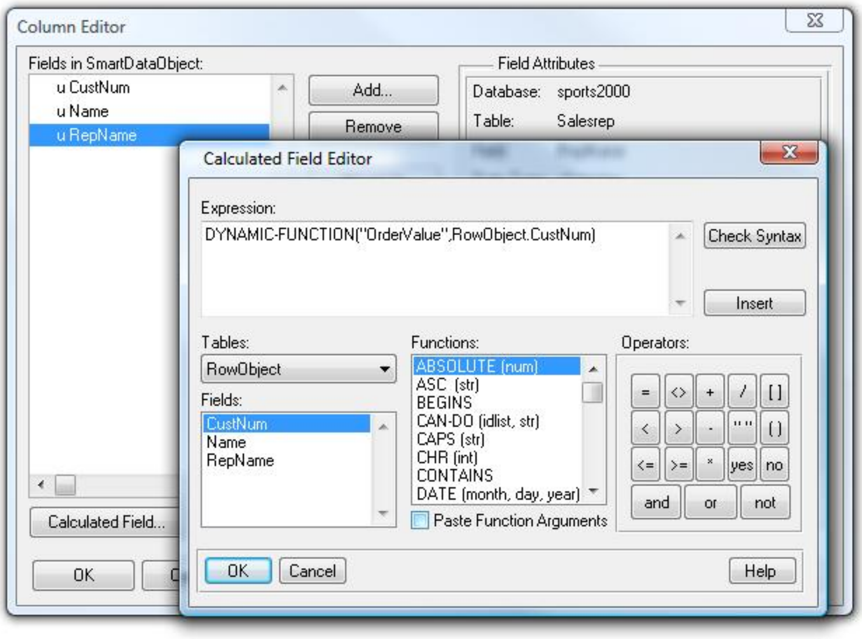


Multi-Field Selector

Available Fields:
Customer.Country
Customer.CreditLimit
Customer.Discount
Customer.EmailAddress
Customer.Fax
Customer.Phone
Customer.PostalCode
Customer.SalesRep
Customer.State
Customer.Terms
Salesrep.MonthQuota[1-12]
Salesrep.Region
Salesrep.SalesRep

Selected Fields:
Customer.CustNum
Customer.Name
Salesrep.RepName

11. Press *Calculated Field* to enter a calculated field.
12. Use the *Calculated Field Editor* to build a calculated field expression and press *OK*.



Column Editor

Fields in SmartDataObject:
u CustNum
u Name
u RepName

Field Attributes
Database: sports2000
Table: Salesrep

Calculated Field Editor

Expression:
DYNAMIC-FUNCTION("OrderValue",RowObject.CustNum)

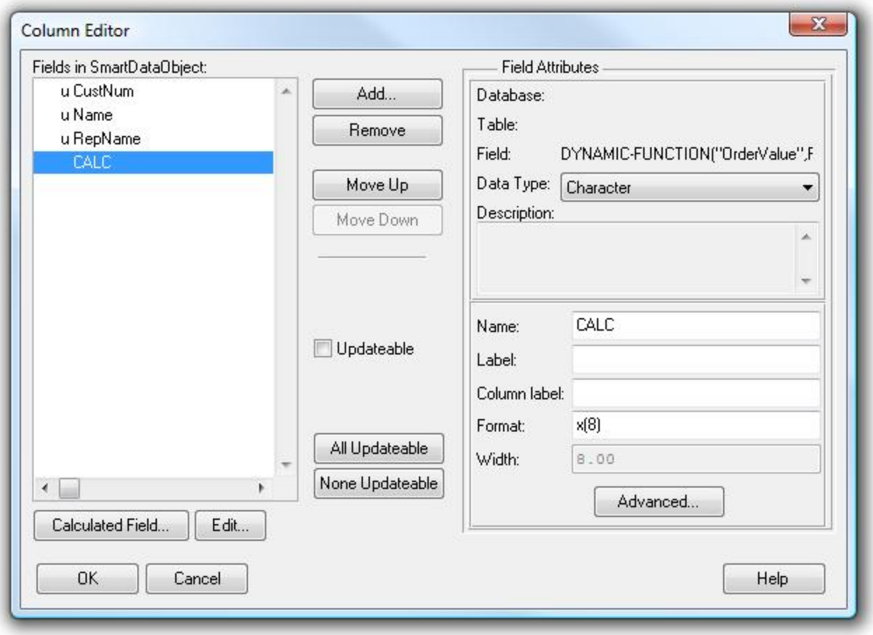
Tables: RowObject

Fields:
CustNum
Name
RepName

Functions:
ABSOLUTE (num)
ASC (str)
BEGINS
CAN-DO (idlist, str)
CAPS (str)
CHR (int)
CONTAINS
DATE (month, day, year)

□ Paste Function Arguments

13. Select the data type and enter a name, label, column label and format for each calculated field. Press OK.



Column Editor

Fields in SmartDataObject:

- u CustNum
- u Name
- u RepName
- CALC

Field Attributes

Database:

Table:

Field: DYNAMIC-FUNCTION("OrderValue",F

Data Type: Character

Description:

Name: CALC

Label:

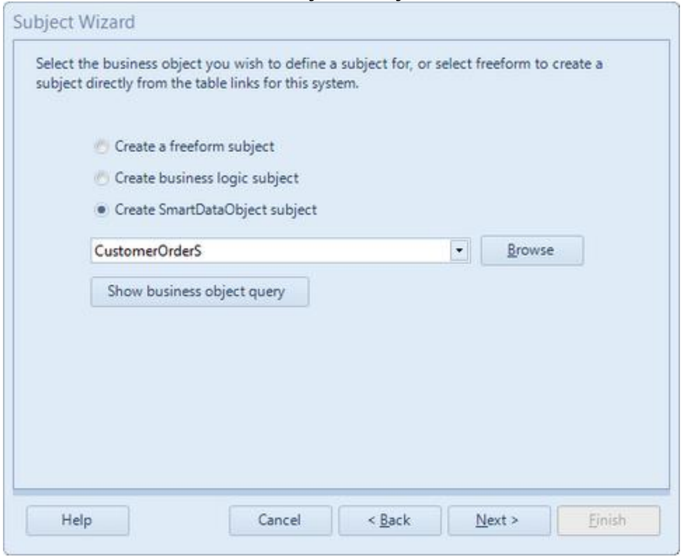
Column label:

Format: x(8)

Width: 8.00

□ Updateable

14. Press *Next* and *Finish*.
15. Save the SDO in the PROPATH of the AppServer.
16. Open the DataPA Analytics Engine screen and ensure your System is selected.
17. Select File → New System
18. Press Next, then enter a name and description for the subject.
19. Select the System and press Next
20. Check Create Smart Data Object Subject, and enter the SDO name. Press Next



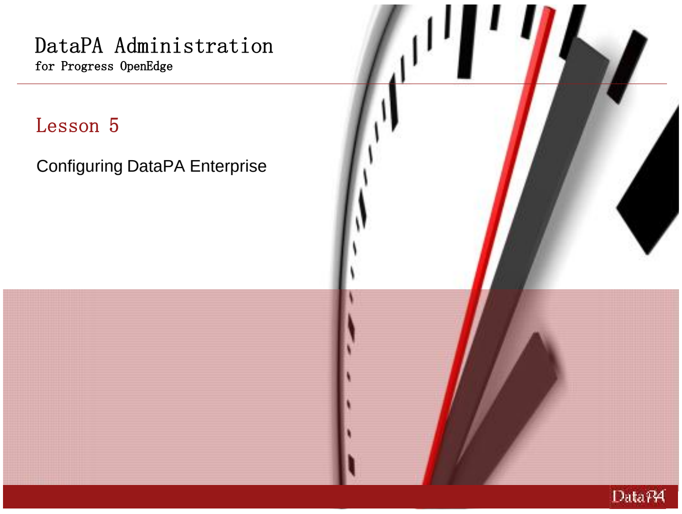
Subject Wizard

Select the business object you wish to define a subject for, or select freeform to create a subject directly from the table links for this system.

○ Create a freeform subject
○ Create business logic subject
● Create SmartDataObject subject

CustomerOrderS

21. Press Next, then enter a name and description for the subject.
22. For each field, enter a column label, and select whether or not it can be used for sorting and conditions.
23. Press Next, press Finish.



DataPA Administration
for Progress OpenEdge

Lesson 5

Configuring DataPA Enterprise

Lesson 5 – Configuring DataPA Enterprise

Introduction

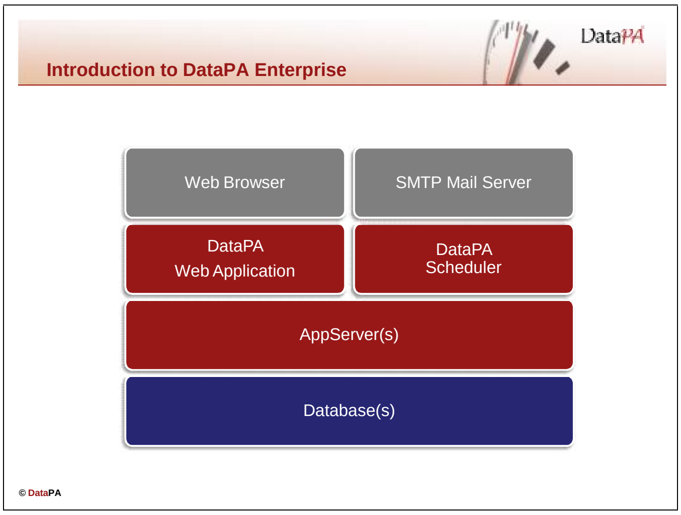
DataPA Enterprise is a server application that allows mobile and browser based access to your live business intelligence and a Scheduler that allows the automated delivery of content via email. With a simple one-click process, any authenticated user can publish content to the server allowing colleagues anywhere in the world instant access. In this lesson you will learn how to configure DataPA Enterprise to suit the needs of your organisation.

Leaning Objectives

When you complete this lesson, you should be able to configure DataPA Enterprise to implement and control content delivery to all stakeholders of your organisation.

Prerequisites

Before you begin this lesson you should be able to create reports, queries and dashboards, install DataPA and configure the DataPA Analytics engine.



Introduction to DataPA Enterprise

DataPA

Web Browser

SMTP Mail Server

DataPA
Web Application

DataPA
Scheduler

AppServer(s)

Database(s)

© DataPA

Introduction to DataPA Enterprise

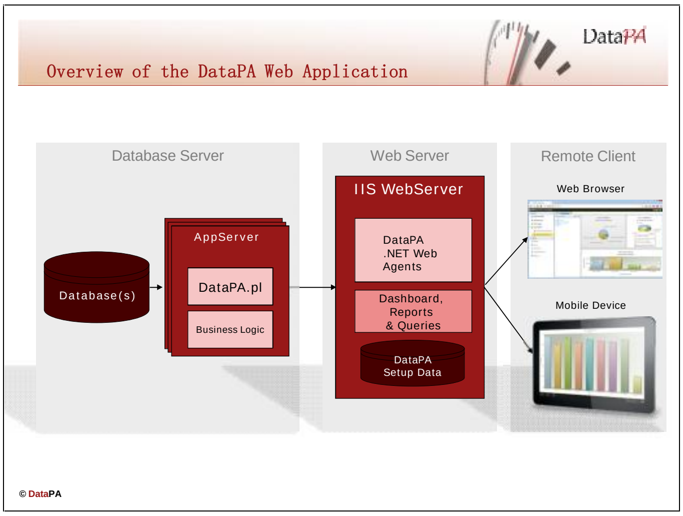
DataPA Enterprise allows Dashboards, Reports and Queries to be easily accessed across a wide area network or the Internet. DataPA Enterprise consists of two key components, the DataPA Web Application, and the DataPA Scheduler.

DataPA Web Application

The DataPA Web Application is a Web Application that allows business users to run dashboard, reports and queries across a wide area network or the Internet, using any standard web browser. Based on Microsoft ASP.NET technology, the DataPA Web Application can be hosted on any compatible IIS Web Server, and connects to your database(s) and runs queries via the AppServer.

DataPA Scheduler

The DataPA Scheduler allows administrators to schedule reports and queries to be run at specific times, and the results emailed to business users, or copied to a particular directory. Report and query results can be saved or emailed on a number of different formats including Adobe Portable Document Format (pdf), Rich Text File (rtf), Excel Spreadsheets and web pages. The DataPA Scheduler can send emails through any appropriately configured SMTP server.



Overview of the DataPA Web Application

DataPA

Database Server

Database(s)

AppServer

DataPA.pl

Business Logic

Web Server

IIS WebServer

DataPA .NET Web Agents

Dashboard, Reports & Queries

DataPA Setup Data

Remote Client

Web Browser

Mobile Device

© DataPA

Introduction

To refresh a dashboard, report or query using the DataPA Web Application three key components are required. These components are usually distinguished by the physical machine on which they reside (although all three could reside on a single machine, or any combination thereof, and the database server could reside across many machines). The three components are the Database server, the Web Server and the Remote Client, and will be described in detail in the following sections.

The Database Server

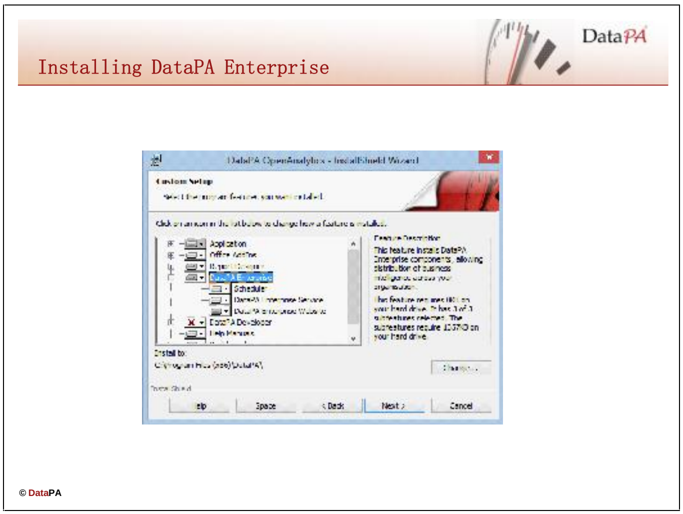
The database server comprises of the database (or databases) and a Progress AppServer configured for DataPA as described in *Lesson 3 – AppServer Administration.* With the exception of an optional business logic procedure to filter reports for the web front end, which will be described later in this lesson, the AppServer required for the DataPA Web Application is configured in exactly the same way as the AppServer for the standard DataPA applications. Indeed, it is often the case that a single AppServer will service both standard and web DataPA users.

The Web Server

The web server comprises of a Microsoft Internet Information Server, with a configured web server, running the DataPA ASP.NET web application. The dashboard, reports, queries and setup data that is required to run these reports should reside on this server. Each Agent in the DataPA ASP.NET Web Application is an instance of DataPA that services client requests, has its own connection to the AppServer and is capable of rendering dashboards, reports and queries for web and mobile devices

The Remote Client

The Remote Client consists of any machine with a browser that supports JavaScript and HTML5 or a mobile device with the relevant DataPA Mobile app installed.



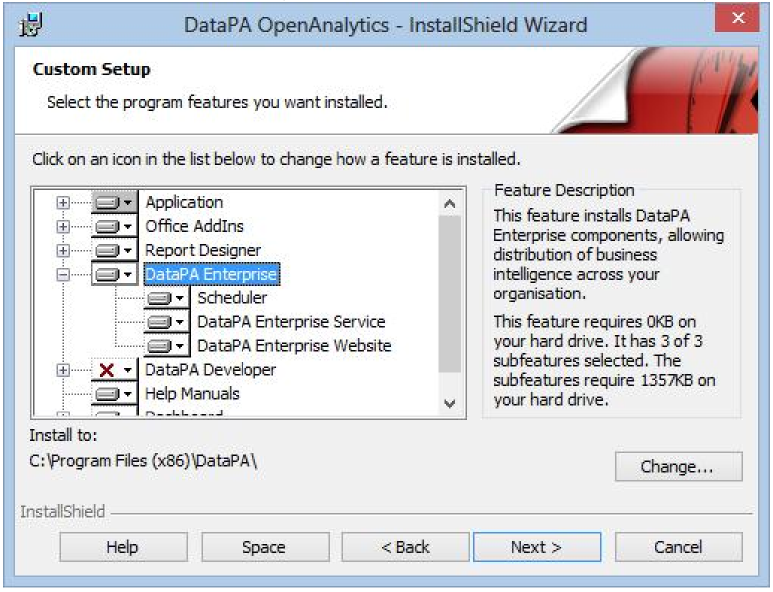
Installing DataPA Enterprise

© DataPA

Installing DataPA Enterprise

To install DataPA Enterprise, follow these steps:

1. Make sure a compatible IIS Web Server is installed on the target machine. DataPA Enterprise supports IIS5, IIS6, IIS7 and IIS8 however we would recommend IIS8 as it offers significant performance advantages.
2. Double click on the DataPA Installation file (setup.exe).
3. Press *Next* when presented with the screen titled *Welcome to the InstallShield Wizard for DataPA*
4. Once you have accepted the license agreement, select the *I accept the terms in the license agreement* option and press *Next*
5. If required, change the installation directory, then press *Next*
6. Enter your name and organisation, select the *Anyone who uses this computer (all users)* option, and press *Next*.
7. Select the *Custom* option, and press *Next*.
8. Make sure the *Scheduler* and *Enterprise* features are set to be installed on the local hard drive. (If the *DataPA Enterprise* feature is not available, check step 1 above).



DataPA OpenAnalytics - InstallShield Wizard

Custom Setup

Select the program features you want installed.

Click on an icon in the list below to change how a feature is installed.

Feature Description

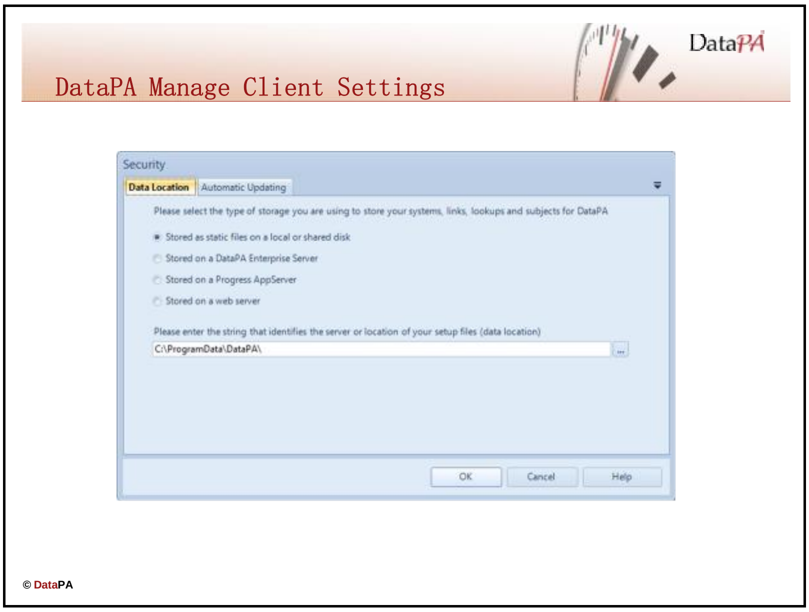
This feature installs DataPA Enterprise components, allowing distribution of business intelligence across your organisation.

This feature requires 0KB on your hard drive. It has 3 of 3 subfeatures selected. The subfeatures require 1357KB on your hard drive.

Install to:

C:\Program Files (x86)\DataPA\

9. Press *Next* then *Install* to complete the installation.



DataPA Manage Client Settings

Security

Data Location | Automatic Updating

Please select the type of storage you are using to store your systems, links, lookups and subjects for DataPA

● Stored as static files on a local or shared disk

○ Stored on a DataPA Enterprise Server

○ Stored on a Progress AppServer

○ Stored on a web server

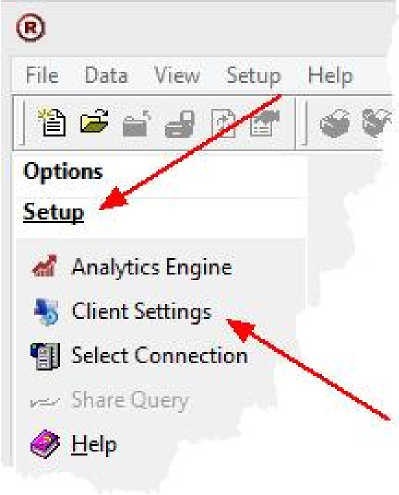
Please enter the string that identifies the server or location of your setup files (data location)

C:\ProgramData\DataPA\

Introduction

The DataPA Client Settings Screen allow the user to control the data location of the Analytics Engine, and of the automatic updating for the client.

Opening the Manage Client Screen - DataPA Reports



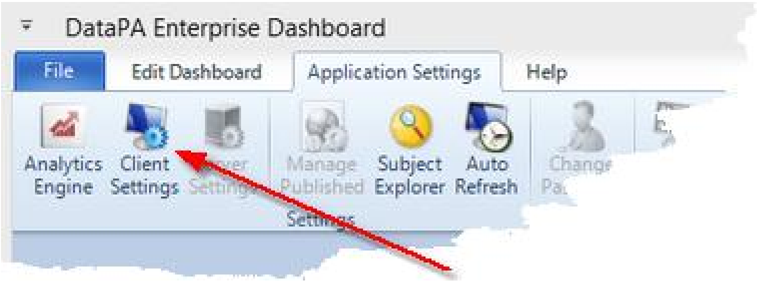
To open the Client settings screen, click on 'Client Settings in the Setup outlook bar (in the main DataPA screen).

If Setup is not visible in the outlook bar, you do not have permissions to change the DataPA Analytics Engine or Client Settings.

If the 'Client Settings' is not visible in the outlook bar, you do not have permissions to change the Client Settings.

Opening the Client Settings Screen - DataPA Enterprise Dashboard

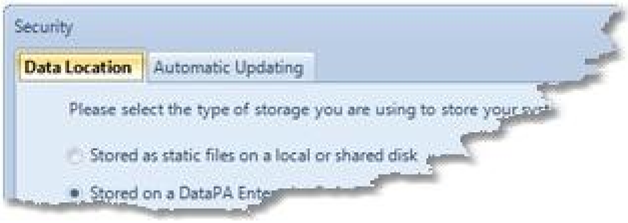
To open the Manage Client screen, click on the *Client Settings* button in the Settings group of the *Application Settings* tab on the ribbon. If the 'Client Settings' button is not visible or enabled, you do not have permissions to change the Client Settings.



Manage Client Settings

Client Settings allows you to control where to store the DataPA Analytics Engine data and determine whether the software will automatically check for updates.

The screen has two tabs, the *Data Location* tab that allows you to control where to store the DataPA system data and the *Automatic Updating* tab that allows you to determine whether the software will automatically check for updates


Security

Data Location | Automatic Updating

Please select the type of storage you are using to store your syst

○ Stored as static files on a local or shared disk

● Stored on a DataPA Ente

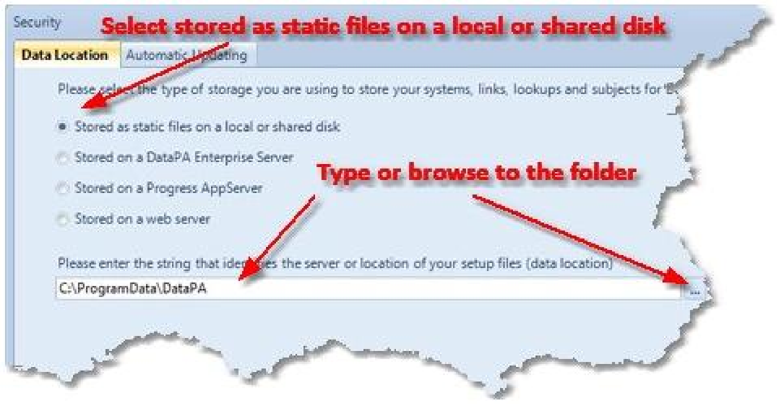
Data Location Tab

The Data Location is the place where all the details of your DataPA setup, the Analytics Engine comprising of the Systems, Links and Subjects are stored and referenced by the software when you run DataPA.

It is often the case that you will want more than one user to share the setup data. As such, the Data Location can be a network drive, a DataPA Enterprise server, a Progress AppServer or a web server.

Data Location - Local or Network Drive

To set your Data Location to a local or network drive, follow these steps;



Select stored as static files on a local or shared disk

Please select the type of storage you are using to store your systems, links, lookups and subjects for D

● Stored as static files on a local or shared disk

○ Stored on a DataPA Enterprise Server

○ Stored on a Progress AppServer

○ Stored on a web server

Type or browse to the folder

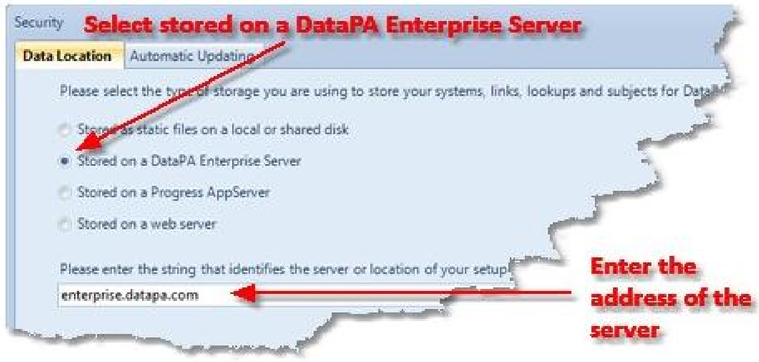
Please enter the string that identifies the server or location of your setup files (data location)

C:\ProgramData\DataPA

1. Select *Stored as static files on a local or shared drive*.

2. Type or browse to the folder where you want the data stored

Data Location - DataPA Enterprise Server



Select stored on a DataPA Enterprise Server

Please select the type of storage you are using to store your systems, links, lookups and subjects for Data

○ Stored as static files on a local or shared disk

● Stored on a DataPA Enterprise Server

○ Stored on a Progress AppServer

○ Stored on a web server

Please enter the string that identifies the server or location of your setup

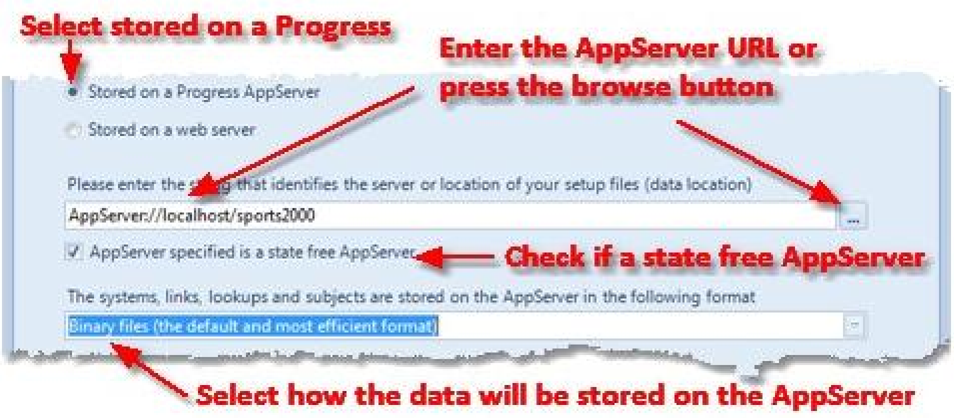
enterprise.datapa.com

Enter the address of the server

1.Select Stored on a DataPA Enterprise server

2. Enter the address of the server. This could be an IP address, or a domain name that identifies the server on the network. For example, if your Enterprise server is installed on a machine that is hosted on the domain www.mydomain.com, enter www.mydomain.com.

Data Location - Progress Appserver



Select stored on a Progress

Enter the AppServer URL or press the browse button

● Stored on a Progress AppServer

○ Stored on a web server

Please enter the string that identifies the server or location of your setup files (data location)

AppServer://localhost/sports2000

☑ AppServer specified is a state free AppServer

Check if a state free AppServer

The systems, links, lookups and subjects are stored on the AppServer in the following format

Binary files (the default and most efficient format)

Select how the data will be stored on the AppServer

1. First you need to copy the setup files you wish to share to the appropriate location. To locate the setup files see your current DataLocation in the Client Settings dialog box. Copy the files Partitions.dat, Links.dat, Lookups.dat, Conditions.dat, Subjects.dat and SDOs.dat from this directory to the working directory of the AppServer.

2. Select Stored on a Progress AppServer

3. Enter the AppServer URL that represents the connection (see below for details), or select the browse button to build it using the AppServer Connection Details window.

4. If the AppServer is a state-free AppServer, check the AppServer specified is a state free AppServer checkbox.

5. If you wish to store the DataPA Analytics Engine data in your database, select the In the database (requires bespoke business logic to store and retrieve) option.

Data Location - Progress Appserver Manual Definition

To connect to the AppServer through the NameServer, use the following syntax:

AppServer://<HOST>[:<PORT NUMBER>]/<APPSERVICE>
[?[username=<USERNAME>][&password=<PASSWORD>] [&appserverinfo=<APPSERVERINFO]]

e.g. AppServer://localhost/sports2000
AppServer://localhost:5162/sports2000
AppServer://localhost/sports2000?username=Nick&password=pwd123
AppServer://localhost/sports2000?appserverinfo=123456

To connect to the AppServer directly, use the following syntax:

AppServerDC://<HOST>:<PORT NUMBER>/<APPSERVICE>
[?[username=<USERNAME>][&password=<PASSWORD>] [&appserverinfo=<APPSERVERINFO]]

e.g. AppServerDC://localhost:5162/sports2000
AppServerDC://localhost:3093/sports2000?username=Nick&password=pwd123

To connect through the Aia, use the standard Progress Aia connection URL.

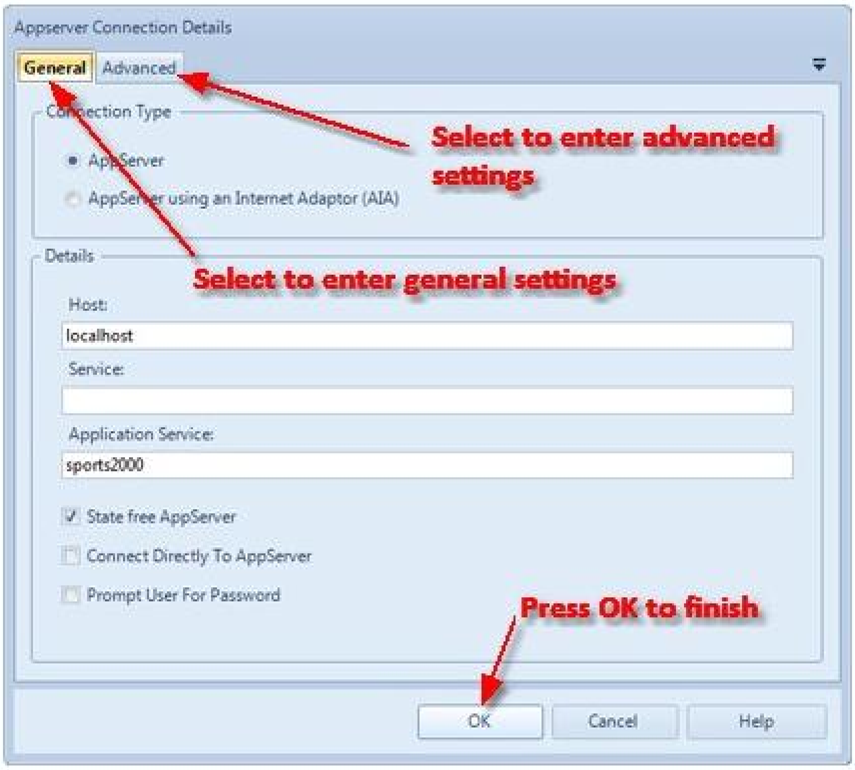
e.g. http://localhost/aia/Aia?AppService=sports2000

Data Location - Progress Appserver Connection Details Screen

The AppServer Connection Details screen (pictured below) allows you to define the connection details for a Progress AppServer connection.

Select the General tab to enter the Main AppServer Connection Details .

Select the Advanced tab to enter Advanced AppServer Connection Details .



Connection Type

● AppServer

○ AppServer using an Internet Adaptor (AIA)

Details

Host: localhost

Service:

Application Service: sports2000

☑ State free AppServer

□ Connect Directly To AppServer

□ Prompt User For Password

Select to enter advanced settings

Select to enter general settings

Press OK to finish

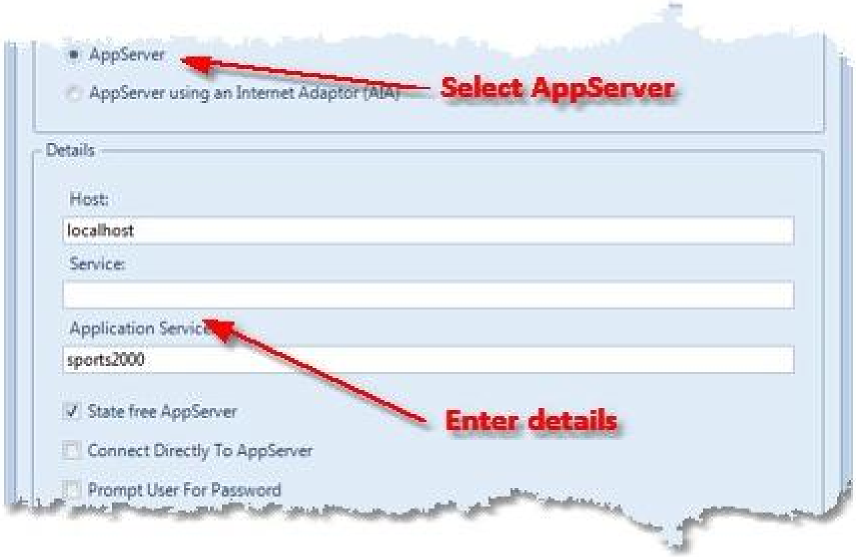
General Tab

There are two types of AppServer connection you can configure. The connection types are listed below with their description. See below for more details on how to configure each connection type.

Connetion Type	Description
Appserver	A direct connection to an AppServer, either through a NameServer or direct to the AppServer broker. Ideal where the AppServer is located on the same local area network (LAN) as the client.
AppServer using an Internet Adapter (AIA)	A remote connection to an AppServer using http tunneling through the Progress AppServer Internet Adaptor. Ideal where the AppServer is located on a public wide area network (WAN) such as the Internet.

Appserver Connections

To configure a Progress AppServer connection, select AppServer from the connection type, then enter the appropriate connection Details (detailed below).



● AppServer

○ AppServer using an Internet Adaptor (AIA)

Details

Host: localhost

Service:

Application Servic sports2000

☑ State free AppServer

□ Connect Directly To AppServer

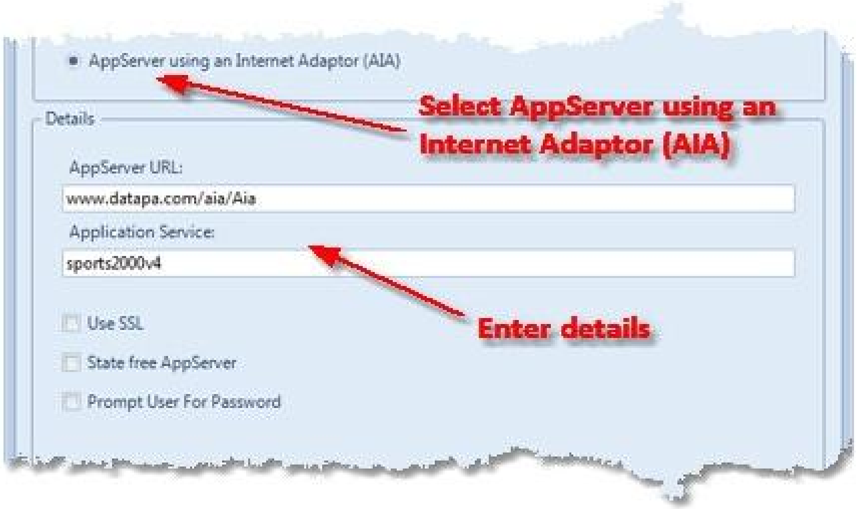
□ Prompt User For Password

Connetion Property	Description
Host	Enter the network name or the IP Address of the machine hosting the Appserver.
Service	Enter the Service Name or Port Number of the Name Sever. If you leave this blank, the default Port umber (5162) will be assumed. If you opt to connect directly to the Appserver rather than through the Name Server, by clicking in the 'Connect Directly To Appserver' check box, you must specify either the Service Name or the Port Number of the Appserver.
Application Service	Enter the Name of the Application Service to connect to.
State free AppServer	Select if you are configuring a connection to an OpenEdge 10 state free appserver (see Progress AppServer Internet Adaptor).
Connect directly to AppServer	Select if you want to connect directly to the AppServer rather than via a NameServer.
Prompt User For Password	Select if you want DataPA to prompt the user for a username and password to validate the connection to the AppServer. The username and password entered by the user will be passed to the AppServer

	connect procedure.
Host	Enter the network name or the IP Address of the machine hosting the Appserver.

Appserver Using an Internet Adapter (AIA) Connection

If you are connecting to the Appserver over the internet, you need to connect via the Appserver Internet Adaptor. To configure a Progress AppServer AIA connection, *AppServer using an Internet Adaptor (AIA)* from the connection type, then enter the appropriate connection *Details* (detailed below).



● AppServer using an Internet Adaptor (AIA)

Details

AppServer URL: www.datapa.com/aia/Aia

Application Service: sports2000v4

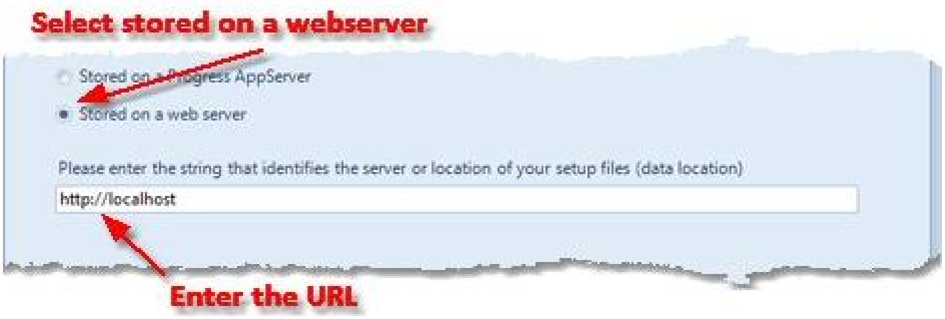
□ Use SSL

□ State free AppServer

□ Prompt User For Password

Connetion Property	Description
AppServer URL	Enter the string which identifies the location of the AIA (Appserver Internet Adaptor). The URL should not include usernames, passwords or http protocol identifiers. e.g. mydomain.com/aia/Aia
Application Service	Enter the Name of the Application Service to connect to.
Use SSL	Select if you wish DataPA to connect to the web server hosting the AIA using SSL encryption
State free AppServer	Select if you are configuring a connection to a state free appserver (see Progress AppServer Internet Adaptor).
Prompt User For Password	Select if you want DataPA to prompt the user for a username and password to validate the connection to the AppServer. The username and password entered by the user will be passed to the AppServer connect procedure.
AppServer URL	Enter the string which identifies the location of the AIA (Appserver Internet Adaptor). The URL should not include usernames, passwords or http protocol identifiers. e.g. mydomain.com/aia/Aia
Application Service	Enter the Name of the Application Service to connect to.

Data Location - Webserver



Select stored on a webserver

○ Stored on a Progress AppServer

● Stored on a web server

Please enter the string that identifies the server or location of your setup files (data location)

http://localhost

Enter the URL

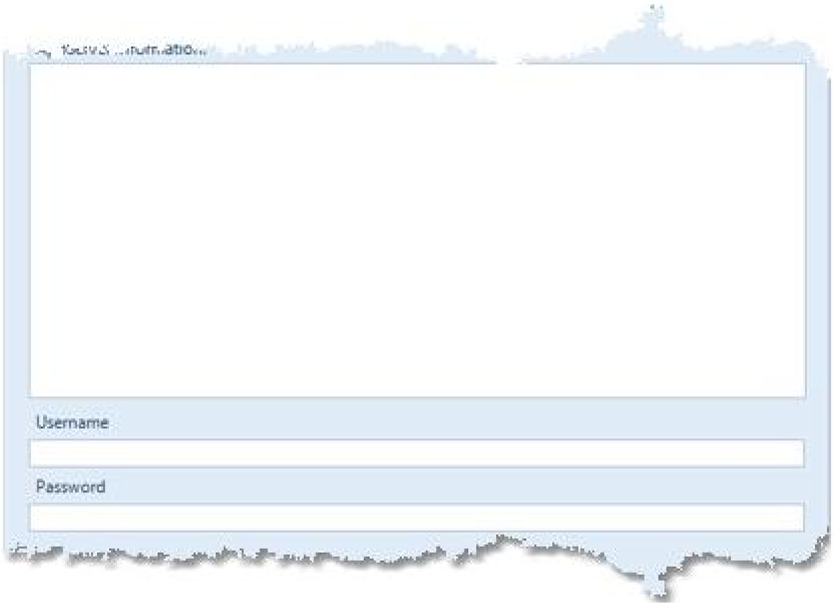
First you need to copy the setup files you wish to share to the appropriate location. To locate the setup files see your current DataLocation. Copy the files Partitions.dat, Links.dat, Lookups.dat, Conditions.dat, Subjects.dat and SDOs.dat from this directory to the web server.

1.Select the Stored on a webserver option.

2.Enter the fully qualified URL of the directory on the webserver

Advanced Tab

As well as passing the Connection Parameters which locate the Appserver, when connecting you can also pass *userid*, *password* and *appserver-info* arguments to the Appserver. These are then passed to the Connect procedure (if defined). If there is no Connect procedure the arguments are discarded.



If you wish to pass appserver-info on connection, type the information that needs to be passed in the 'Appserver Information' text box.

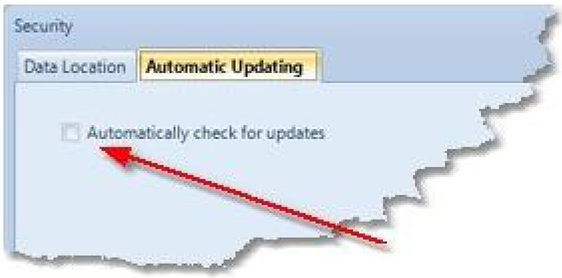
If you wish to pass a username on connection, type the information that needs to be passed in the 'Username' text box.

If you wish to pass a password on connection, type the information that needs to be passed in the 'Password' text box.

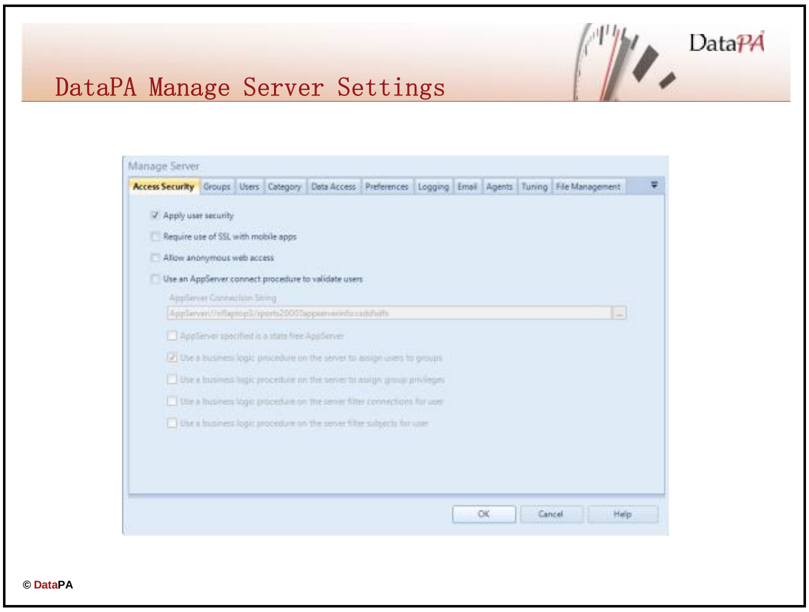
Automatic Updating Tab

By default, DataPA OpenAnalytics will check online for any available updates when the applications are opened.

If you would like to disable the automatic update functionality, uncheck the Automatic check for updates option.



□ Automatically check for updates



DataPA Manage Server Settings

Manage Server

☑ Apply user security

□ Require use of SSL with mobile apps

□ Allow anonymous web access

□ Use an AppServer connect procedure to validate users

AppServer Connection String

AppServer://nflaptop3/sports2000?appserverinfo=sdsfsdfs

□ AppServer specified is a state free AppServer

☑ Use a business logic procedure on the server to assign users to groups

□ Use a business logic procedure on the server to assign group privileges

□ Use a business logic procedure on the server filter connections for user

□ Use a business logic procedure on the server filter subjects for user

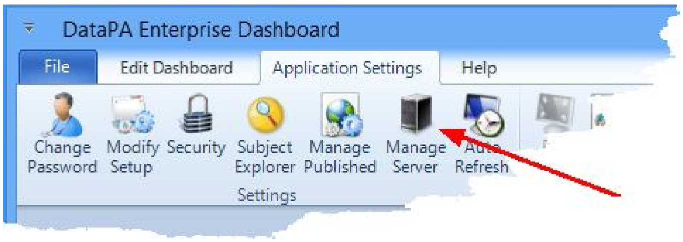
Introduction

The DataPA Manage Server settings allow control setup data locking and location and configure automatic updating for users on a particular machine. These settings are designed for organizations who wish to use Microsoft Domain security to control the client behaviour of DataPA.

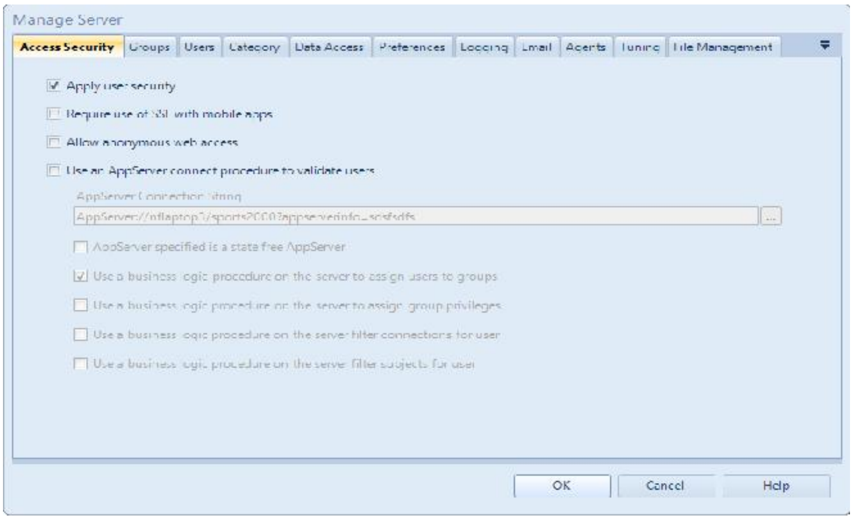
DataPA Enterprise is configured from the Manage Server screen in the client applications. If, in the *Security* options, you elect to set your *Data Location to a DataPA Enterprise Server* then the Manage Server screen becomes available. The manage server screen, access to which is controlled via a *group permission item,* allows the administrator to manage all aspects of the behaviour of the DataPA Enterprise processes.

Opening the Manage Server Screen

To open the Manage Server screen, click on the *Security* button in the Settings group of the *Application Settings* tab on the ribbon. If the 'Manage Server' button is not visible or enabled, you do not have permissions to change the Server Settings.



Manage Server Settings



☑ Apply user security

□ Require use of SSL with mobile apps

□ Allow anonymous web access

□ Use an AppServer connect procedure to validate users

AppServer Connection String

AppServer://nflaptop1/sports2000?appserverinfo=sdsfsdfs

□ AppServer specified is a state free AppServer

☑ Use a business logic procedure on the server to assign users to groups

□ Use a business logic procedure on the server to assign group privileges

□ Use a business logic procedure on the server filter connections for user

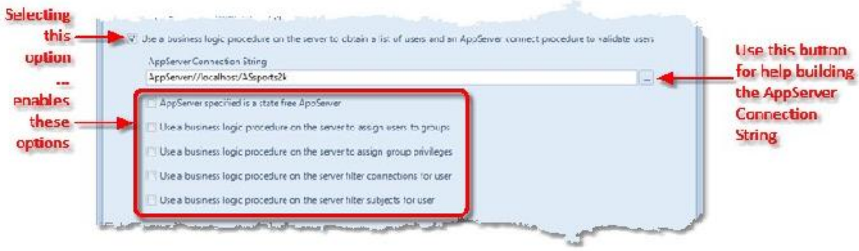
□ Use a business logic procedure on the server filter subjects for user

The screen comprises of the following tabs

Tabs	Description
Access Security	Allows you to set high-level security options relating to the DataPA Enterprise Service
Groups	Allows you to administer groups, to which users belong, that control access to functions and content.
Users	Allows you to administer users that have access to DataPA Enterprise.
Category	Allows you to administer categories to which content is published and access can be granted or denied to specific groups.
Data Access	Allows you to administer security details DataPA Enterprise will use when connecting to a business application.
Preferences	Allows you to administer the look and feel of the DataPA Enterprise we application.
Logging	Allows you to administer different logging methods on the DataPA Enterprise processes.
Email	Allows you to administer email alerts to notify administrators when key events occur.
Agents	Allows you to administer the use of agents within DataPA Enterprise.
Tuning	Allows you to tune memory management within DataPA Enterprise.
File Management	Allows you to administer file locations and limits for data stored by DataPA Enterprise.

Access Security

This screen is used to set high-level security options relating to the DataPA Enterprise Service. The available options are:

Option	Description
Apply user security	Set this option to enable user security. Once set, users need to log-in upon starting DataPA client applications and before accessing content via the DataPA Enterprise Service.
Require use of SSL with mobile apps	This option enables a security feature which requires that any mobile apps connecting to the DataPA Enterprise Service are secured by the use of encryption.
Use a business logic procedure on the server to obtain a list of users and an AppServer connect procedure to validate users	This option allows you to use your own business logic to provide a list of users for the Security setup. For example, your application itself may be the source of truth for your list of users, using this option will then save you having to enter them all again in DataPA. To use this feature you MUST have a program named PAGetUsers.p available to the specified AppServer (i.e. in it's PROPATH). An initial version of this can be found in the "Server Procedure Examples" folder located in the "DataPA OpenAnalytics" directory created under your Start menu by the DataPA installation. Modify to populate the list of users as required. To validate users' passwords you can specify a connect procedure on your AppServer - please see AppServer Connect Procedure.  ☑ Use a business logic procedure on the server to obtain a list of users and an AppServer connect procedure to validate users AppServer Connection String: AppServer//localhost/ASsports2k □ AppServer specified is a state free AppServer □ Use a business logic procedure on the server to assign users to groups □ Use a business logic procedure on the server to assign group privileges □ Use a business logic procedure on the server filter connections for user □ Use a business logic procedure on the server filter subjects for user Selecting this option ... enables these options Use this button for help building the AppServer Connection String
AppServer Connection String	If you set the preceding option you will be required to enter an AppServer connection string. You can use the "..." button to automatically construct the string with the correct syntax.
AppServer specified is a state free AppServer	Select this option if the AppServer specified above is running in state free mode.
Use a business logic procedure on the server to assign users to groups	This option allows you to use your own business logic to assign users to group. You may do this if your application already has some kind of data on which to base this decision. To use this feature you ***MUST*** have a program named PAUserGroup.p available to the specified AppServer (i.e. in its PROPATH). An initial version of this can be found in the "Server Procedure Examples" folder located in the "DataPA OpenAnalytics" directory created under your Start menu by the DataPA installation. Modify to assign users to groups as required. Notes:

	• If the group assigned to the user by your business logic does not exist, the user will be assigned to the default group instead. • If this option is enabled the "Users" tab of the Manage Server window will no longer be visible; as it is no longer needed. **NOTE:** Any existing users you have set up, with the exception of the admin user, will be lost as the list of users is overwritten with those retrieved from the server.
Use a business logic procedure on the server to assign group privileges	This option allows you to use your own business logic to assign permission items available to a group. You may do this if your application already has data of some kind on which to base this decision. The permission items you assign in this way are identical in function to those specified on the *Groups* tab of the Manage server window. To use this feature you ***MUST*** have a program named PAGroupPrivileges.p available to the specified AppServer (i.e. in it's PROPATH). An initial version of this can be found in the "Server Procedure Examples" folder located in the "DataPA OpenAnalytics" directory created under your Start menu by the DataPA installation.
Use a business logic procedure on the server filter connections for user	This option allows you to control which users will have access to which *connections* for a given system. For example, a software developer may only be allowed to connect to a Test system, but not Production. To use this feature you ***MUST*** have a program named PAFilterConnections.p available to the specified AppServer (i.e. in it's PROPATH). An initial version of this can be found in the "Server Procedure Examples" folder located in the "DataPA OpenAnalytics" directory created under your Start menu by the DataPA installation. DataPA passes a temp-table of connections to this procedure, along with the current user's ID. Modify the procedure to delete any connections to which the user should not have access.
Use a business logic procedure on the server filter subjects for user	This option allows you to control which users will have access to which *subjects.* Subjects are essentially data views and are the building blocks from which DataPA queries are constructed. To use this feature you ***MUST*** have a program named PAFilterSubjects.p available to the specified AppServer (i.e. in it's PROPATH). An initial version of this can be found in the "Server Procedure Examples" folder located in the "DataPA OpenAnalytics" directory created under your Start menu by the DataPA installation. DataPA passes a temp-table of subjects to this procedure, along with the current user's ID. Modify the procedure to delete any subjects to which the user should not have access.

Server Side Procedures

Get Users from Server

The server procedure to get a list of users from your business application must be called PAGetUsers.p and reside in the PROPATH of the AppServer.

The procedure passes back a temp-table with single character field containing the list of user names.

The definitions for this procedure should be as follows:

```
DEFINE TEMP-TABLE ttUsers
  FIELD cUserName as CHARACTER.

DEFINE OUTPUT PARAMETER TABLE FOR ttUsers.
```

An example of the body code for PAgetUsers.p is as follows:

```
CREATE ttUsers.
ASSIGN ttUsers.cUsername = "TerstUser1".
```

Assign Users to Groups

The server procedure to assign a user to a group from your business application must be called PAUserGroup.p and reside in the PROPATH of the AppServer.

The procedure receives the username as an input parameter, and returns a group name as an output parameter. The username received by PAUserGroup.p is the username entered by the user when DataPA connects to the AppServer, if available, otherwise the users Windows username. This username can be used to determine the security settings that should be returned.

The definitions for this procedure should be as follows:

```
DEFINE INPUT PARAMETER ip-cUsername AS character NO-undo.
DEFINE OUTPUT PARAMETER op-cGroup AS character NO-undo.
```

An example of the body code for PAUserGroup.p is as follows:

```
CASE ip-cUserName:
  WHEN "Gary" THEN DO:
    op-cGroup = "Administrator".
  END.
  WHEN "Nick" THEN DO:
    op-cGroup = "Default".
  END.
  WHEN "mfgadmin" THEN DO:
    op-cGroup = "Administrator".
  END.
END.
```

Assign Group Privileges

The server procedure to assign privileges to a group from your business application must be called PAGroupPrivileges.p and reside in the PROPATH of the AppServer.

The procedure receives a ttGroups temp table as an input parameter, and returns a temp table ttGroupPrivileges as an output parameter. The definitions for this procedure should be as follows:

```
DEFINE TEMP-TABLE ttGroups
  FIELD cGroupName      AS CHARACTER.

DEFINE TEMP-TABLE ttGroupPrivileges
  FIELD cGroupName              AS CHARACTER
  FIELD AllowLinks              AS LOGICAL
  FIELD AllowSecurity           AS LOGICAL
  FIELD AllowSetup              AS LOGICAL
  FIELD AllowSubject            AS LOGICAL
  FIELD AllowSystem             AS LOGICAL
  FIELD AllowSetupLockOverride  AS LOGICAL
  FIELD UserLevel               AS CHARACTER
  FIELD RegUser                 AS CHARACTER
  FIELD RegOrganisation         AS CHARACTER
  FIELD RegSerialNum            AS CHARACTER
  FIELD RegCode                 AS CHARACTER
  FIELD Administrator           AS LOGICAL
  FIELD AllowPublish            AS LOGICAL
  FIELD RequireSecurityPassword AS LOGICAL
  FIELD SecurityPassword        AS CHARACTER.

DEFINE INPUT  PARAMETER TABLE FOR ttGroups.
DEFINE OUTPUT PARAMETER TABLE FOR ttGroupPrivileges.
```

The fields in the ttGroupPrivileges temp table are as follows:

Field	Description
cGroupName	The group name that the permissions relate to.
AllowLinks	Indicates whether or not the user will be able to create, delete and modify links. 0=False,1=True,2=Default
AllowSecurity	Indicates whether or not the user will be able to modify security settings for that session. 0=False,1=True,2=Default
AllowSetup	Indicates whether or not the user will have access to the setup screen. 0=False,1=True,2=Default
AllowSubject	Indicates whether or not the user will be able to create, delete and modify subjects. 0=False,1=True,2=Default
AllowSystem	Indicates whether or not the user will be able to create, delete and modify Systems. 0=False,1=True,2=Default
AllowSetupLockOverride	Set to True to allow the user to override the lock for themselves or another user on setup screen. This will result in the first user being unable to save any changes to the setup files. (Version 3.00.0064 and above only)
UserLevel	If set to Query, will prevent the user from creating or modifying queries and reports.
RegUser	If the client is not licensed, the user name for the client license to apply.

RegOrganisation	If the client is not licensed, the organization for the client license to apply.
RegSerialNum	If the client is not licensed, the serial number for the client license to apply.
RegCode	If the client is not licensed, the registration code for the client license to apply.
Administrator	Indicates whether or not the user is an Administrator. 0=False,1=True,2=Default
Allow Publish	Indicates whether or not the user will be able to publish content to the DataPA Enterprise Service. 0=False,1=True,2=Default
RequireSecurityPassword	Indicates whether or not the user requires a password to access the security screen. 0=False,1=True,2=Default
SecurityPassword	The password required to access the security screen if required.

An example of the body code for PAGroupPrivileges.p is as follows:

```
FOR EACH ttGroups:
 CASE ttGroups.cGroupName:
   WHEN "Administrator" THEN DO:
     CREATE ttGroupPrivileges.
     ASSIGN ttGroupPrivileges.cGroupName       = ttGroups.cGroupName
            ttGroupPrivileges.AllowLinks        = TRUE
            ttGroupPrivileges.AllowSecurity     = TRUE
            ttGroupPrivileges.AllowSetup        = TRUE
            ttGroupPrivileges.AllowSubject      = TRUE
            ttGroupPrivileges.AllowSystem        = TRUE
            ttGroupPrivileges.AllowSetupLockOverride  = TRUE
            ttGroupPrivileges.UserLevel         = "OpenAnalytics Developer"
            ttGroupPrivileges.RegUser           = "DataPA Limited"
            ttGroupPrivileges.RegOrganisation   = "DataPA Limited"
            ttGroupPrivileges.RegSerialNum      = "0123456789"
            ttGroupPrivileges.RegCode           = "E9DB60908290D98191E8723B"
            ttGroupPrivileges.Administrator     = TRUE
            ttGroupPrivileges.AllowPublish      = TRUE
            ttGroupPrivileges.RequireSecurityPassword = TRUE
            ttGroupPrivileges.SecurityPassword  = "abc123".
   END.
   WHEN "Default" THEN DO:
     CREATE ttGroupPrivileges.
     ASSIGN ttGroupPrivileges.cGroupName       = ttGroups.cGroupName
            ttGroupPrivileges.AllowLinks        = FALSE
            ttGroupPrivileges.AllowSecurity     = FALSE
            ttGroupPrivileges.AllowSetup        = FALSE
            ttGroupPrivileges.AllowSubject      = FALSE
            ttGroupPrivileges.AllowSystem       = FALSE
            ttGroupPrivileges.AllowSetupLockOverride  = FALSE
            ttGroupPrivileges.UserLevel         = "OpenAnalytics Business"
            ttGroupPrivileges.RegUser           = "DataPA Limited"
            ttGroupPrivileges.RegOrganisation   = "DataPA Limited"
            ttGroupPrivileges.RegSerialNum      = "9876543210"
            ttGroupPrivileges.RegCode           = "8455C08157304ECAC1F304A7"
            ttGroupPrivileges.Administrator     = TRUE
            ttGroupPrivileges.AllowPublish      = TRUE
            ttGroupPrivileges.RequireSecurityPassword = TRUE
            ttGroupPrivileges.SecurityPassword  = "abc123".
   END.
 END CASE.
END.
```

Filter Connections for User

You may want to limit the systems or connections available to particular users, so they can only open and run queries or reports against a certain system, or connection.

The server procedure to filter connections for a user from your business application must be called PAFilterConnections.p and reside in the PROPATH of the AppServer.

The procedure passes the username as an input parameter and Progress temp-table as in input-output parameter that contains all the connections. If any of the records are deleted from that temp-table, the corresponding connection will not be available to the user.

The definitions for this procedure should be as follows:

```
DEFINE TEMP-TABLE ttConnections
  FIELD cName           AS CHARACTER
  FIELD cConnectionName AS CHARACTER
  FIELD bPrimary        AS LOGICAL
  FIELD bAdmin          AS LOGICAL.

DEFINE INPUT  PARAMETER ip-cUserName     AS CHARACTER NO-UNDO.
DEFINE INPUT-OUTPUT PARAMETER TABLE FOR ttConnections.
```

The username received by PAFilterConnnection is the username entered by the user when DataPA connects to the AppServer, if available, otherwise the users Windows username. The procedure can then delete connections you do not want to be available to a user. An example of the body code for PAFilterConnnection.p is as follows:

```
DEFINE VARIABLE cGroups AS CHARACTER   NO-UNDO.

RUN GetUserGroup(INPUT ip-cUserName, OUTPUT cGroups).
FOR EACH ttConnections:
  IF NOT CAN-DO("Finance", cGroups) THEN DO:
    IF cName = "Finance" THEN DELETE ttConnections.
  END.
END.
```

Filter Subjects for User

You may want to limit the subjects for a particular user, so they can only open and run queries or reports against a certain system, or connection for which they have permission.

The server procedure to filter subjects for a user from your business application must be called PAFilterSubjects.p and reside in the PROPATH of the AppServer.

The procedure passes the username and two Progress input-output parameter temp-tables containing all the subjects (ttSubjects) and their corresponding fields (ttSubjectFields). If any of the subject records are deleted, the subject will not be available to the user and if any of the fields are deleted, the corresponding subject field will not be available to the end user.

The definitions for this procedure should be as follows:

```
DEFINE TEMP-TABLE ttSubjects NO-UNDO
  FIELD cID              AS CHARACTER
  FIELD cTitle           AS CHARACTER
  FIELD cSystemName      AS CHARACTER
  FIELD cDescription     AS CHARACTER
  FIELD cSmartDataObject AS CHARACTER
  FIELD lDynSDO          AS LOGICAL
  FIELD cTables          AS CHARACTER
  FIELD cParents         AS CHARACTER
  FIELD cBuffers         AS CHARACTER
  FIELD cIndexes         AS CHARACTER
  FIELD cLinks           AS CHARACTER
  FIELD bDynamic         AS LOGICAL FORMAT "true/false"
  INDEX i1 AS PRIMARY UNIQUE cID.

DEFINE TEMP-TABLE ttSubjectFields NO-UNDO
  FIELD cID            AS CHARACTER
  FIELD cTitle         AS CHARACTER
  FIELD cFieldName     AS CHARACTER
  FIELD cDataType      AS CHARACTER
  FIELD cLabel         AS CHARACTER
  FIELD cFormat        AS CHARACTER
  FIELD lAllowIndex    AS LOGICAL
  FIELD lAllowSort     AS LOGICAL
  FIELD lAllowContains AS LOGICAL
  FIELD iExtent        AS INTEGER
  FIELD iWidth         AS INTEGER
  FIELD cExpression    AS CHARACTER
  FIELD cSvrFormat     AS CHARACTER
  FIELD cLookup        AS CHARACTER
  INDEX i1 AS PRIMARY UNIQUE cID cFieldName.

DEFINE INPUT  PARAMETER ipcUserName AS CHARACTER NO-UNDO.
DEFINE OUTPUT PARAMETER opcUseDir   AS CHARACTER NO-UNDO.
DEFINE INPUT-OUTPUT PARAMETER TABLE FOR ttSubjects.
DEFINE INPUT-OUTPUT PARAMETER TABLE FOR ttSubjectFields.
```

The username received by PAFilterSubjects is the username entered by the user when DataPA connects to the AppServer, if available, otherwise the users Windows username.

The ttSubjects temp-table contains all the subjects available to the user, and the ttSubjectFields temp-table contains all the fields for each subject joined by the primary index. If you delete any ttSubjects records, the corresponding subject will not be available to the user. If you delete any ttSubjectFields records, those fields will not be available to the end user if they use the subject.

An example of the body code for PAFilterConnnection.p is as follows:

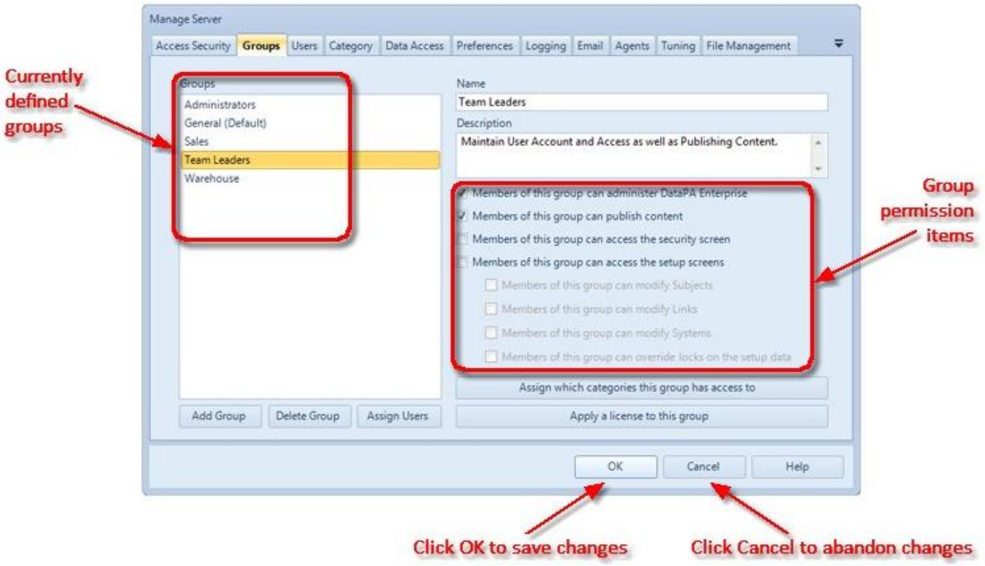
```
DEFINE VARIABLE lFound AS LOGICAL     NO-UNDO.
DEFINE VARIABLE cGroups AS CHARACTER   NO-UNDO.

RUN GetUserGroup(INPUT ipcUserName, OUTPUT cGroups).
FOR EACH ttSubjects:
  lFound = FALSE.
  FOR EACH ttSubjectFields OF ttSubjects:
    IF NOT CAN-DO("Finance", cGroups)
    AND ttSubjectFields.cFieldName = "sports2000.Customer.Balance"
    THEN lFound = TRUE.
  END.
  IF lFound THEN DELETE ttSubjects.
END.
```

Groups

The security model for DataPA Enterprise - by which users can access content - uses the concept of groups...

- Content is published to Categories
- Users belong to a Group
- Groups are granted access to Categories

The "Groups" tab of the Manage Server screen is intended to allow maintenance of these groups



Groups
Administrators
General (Default)
Sales
Team Leaders
Warehouse

Name: Team Leaders

Description: Maintain User Account and Access as well as Publishing Content.

☑ Members of this group can administer DataPA Enterprise
☑ Members of this group can publish content
□ Members of this group can access the security screen
□ Members of this group can access the setup screens
□ Members of this group can modify Subjects
□ Members of this group can modify Links
□ Members of this group can modify Systems
□ Members of this group can override locks on the setup data

Currently defined groups

Group permission items

Click OK to save changes

Click Cancel to abandon changes

The Administrators and Default groups are created automatically by DataPA. As different DataPA licenses give different functionality in the application, you can configure a particular group to use a specific license.

Default Groups

Group	Description
Administrators	The administrators group has full permissions and cannot be removed, although it ***may*** be renamed. By default there is a user - admin - configured as a member of this group. The initial password for this user is "admin". The idea of having an admin user belonging to the administrator group which cannot be removed is to make it impossible to get into a situation where the security setup can no longer be maintained. Although the admin user cannot be removed, the admin password ***can*** be changed to prevent other users accessing the account.

Default	Similarly to the administrators group, the default group cannot be removed, although it ***may*** be renamed. The text "Default" is appended to the name in parenthesis so that even if it is renamed it is still apparent that it is the default group. Unlike the administrators group, the default group can have it's permissions items changed. The purpose of the default group is to catch users for whom the system cannot determine to which group they belong. Such a situation may arise if ● The group to which a user belongs is deleted, or ● *Business logic is used to assign users to groups*, and the business logic returns an unknown group.

There are two groups created automatically by DataPA which cannot be removed:

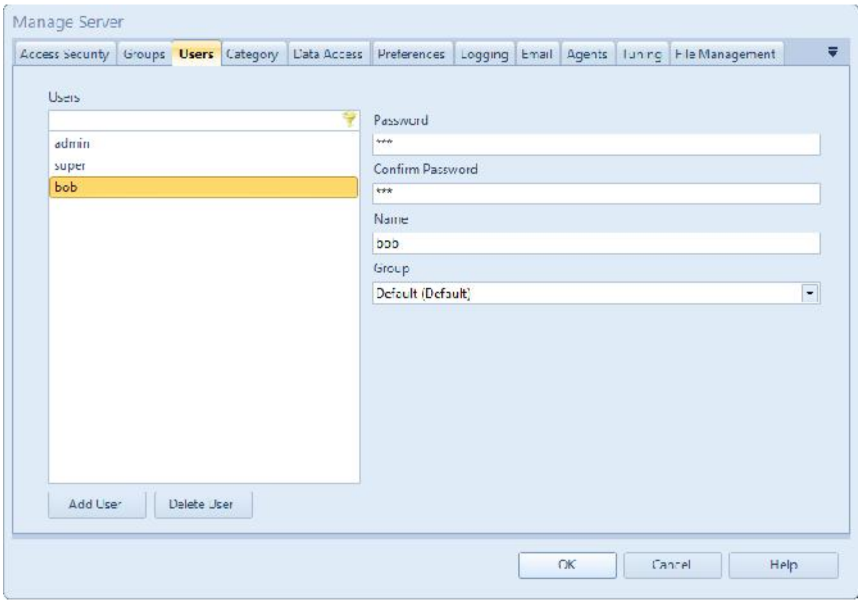
Default Groups

The following permission items are granted at the group level:

Permission	Description
Members of this group can administer DataPA Enterprise	This permission item specifies whether members of the selected group can access and make changes in the *Manage Server* window. This window is used to maintain security, groups categories and users
Members of this group can publish content.	This permission item specifies whether members of the selected group can publish content in the Manage Published window. This window is used to control content which has been published to the DataPA Enterprise Service.
Members of this group can access the Client Setup Screen	This permission item specifies whether members of the selected group can access and make changes in the *Manage Client* window. This window is used to control where to store the DataPA CME system data and determine whether the software will automatically check for updates
Members of this group can access the Setup screens	This permission item specifies whether members of the selected group can access and make changes in the *Modify SetUp* window. This window is used to maintain details of *Systems*, *Links* and *Subjects* for DataPA.
Members of this group can modify Subjects	This permission item specifies whether members of the selected group can add amend and delete *Subjects* via the *DataPA Analytics Engine* screen.
Members of this group can modify Links	This permission item specifies whether members of the selected group can add amend and delete *Links* via the *DataPA Analytics Engine* screen.
Members of this group can modify Systems	This permission item specifies whether members of the selected group can add amend and delete *Systems* via the *DataPA Analytics Engine* screen
Members of this group can override locks on the setup data	Whenever a user enters the *DataPA Analytics Engine* screen, DataPA creates a lock file. The purpose of this lock file is to prevent separate users attempting to make changes to the setup simultaneously. On rare occasions this lock file may be left behind by an interrupted session. This permission item allows members of the selected group to continue to make changes to the setup even if such a lock file exists.

Users

The "Users" tab of the Manage Server screen is intended to allow maintenance of users.



Users
admin
super
bob

Password
***

Confirm Password
***

Name
bob

Group
Default (Default)

Users have a login ID, name and password.

They may also be assigned to a group here.

Note a user - "admin" - is automatically created by DataPA and cannot be deleted. The initial password for this user is "admin". Similarly an "Administrators" group, which has full permissions and cannot be removed is also created. The admin user is automatically configured as a member of this group and this cannot be changed.

The idea of having an admin user belonging to the administrator group which cannot be removed is to make it impossible to get into a situation where the security setup can no longer be maintained.

Although the admin user cannot be removed, the admin password ***can*** be changed to prevent other users accessing the account.

Category

The security model for DataPA Enterprise - by which users can access content - uses the concept of categories...

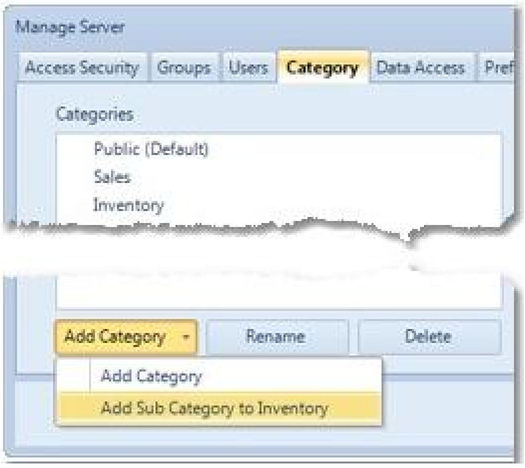
Content is published to Categories

- Users belong to a Group
- Groups are granted access to Categories

The "Category" tab of the Manage Server screen is intended to allow maintenance of these categories.

They can be added, renamed or deleted, and the groups which have access to each category may also be set here.

Note that you can add sub-categories to categories to create a hierarchical structure...



Data Access

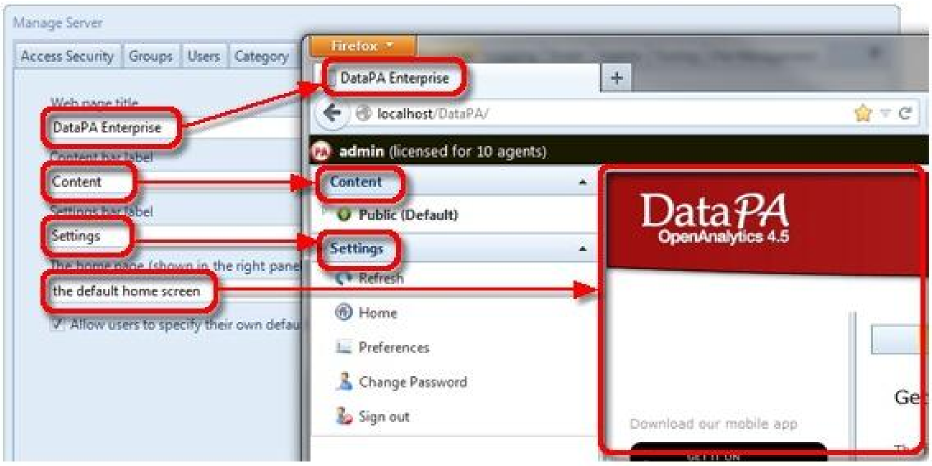
When using the DataPA Enterprise Service to create shared content, it is often useful to set up a refresh schedule so that the content is refreshed appropriately.

However, you may have configured your *AppServer to require a username and password* when connecting.

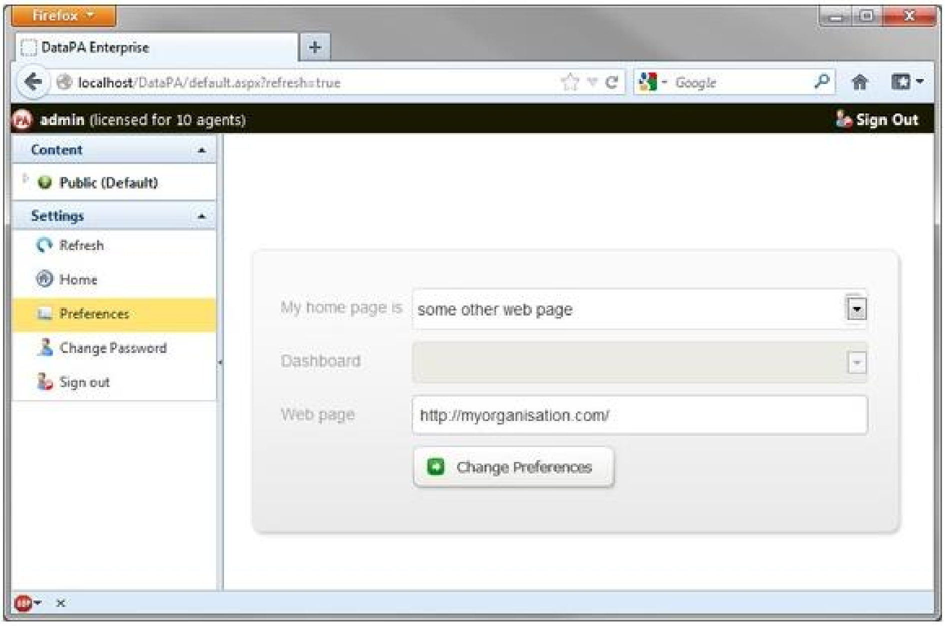
On this tab of the Manage Server window you can set the credentials to be used by the service when connecting to the AppServer to perform this refresh, unless a specific username and password has been defined in the *refresh schedule screen* for a particular query.

Preferences

The preferences tab of the manage server window allows you to control the look and feel of your DataPA Enterprise web page as illustrated below...


Web page title: DataPA Enterprise
Content bar label: Content
Settings bar label: Settings
The home page (shown in the right pane: the default home screen
☑ Allow users to specify their own defau

If you opt to allow users to specify their own default home page they do so via a preferences option on the Enterprise web page...



My home page is: some other web page

Dashboard:

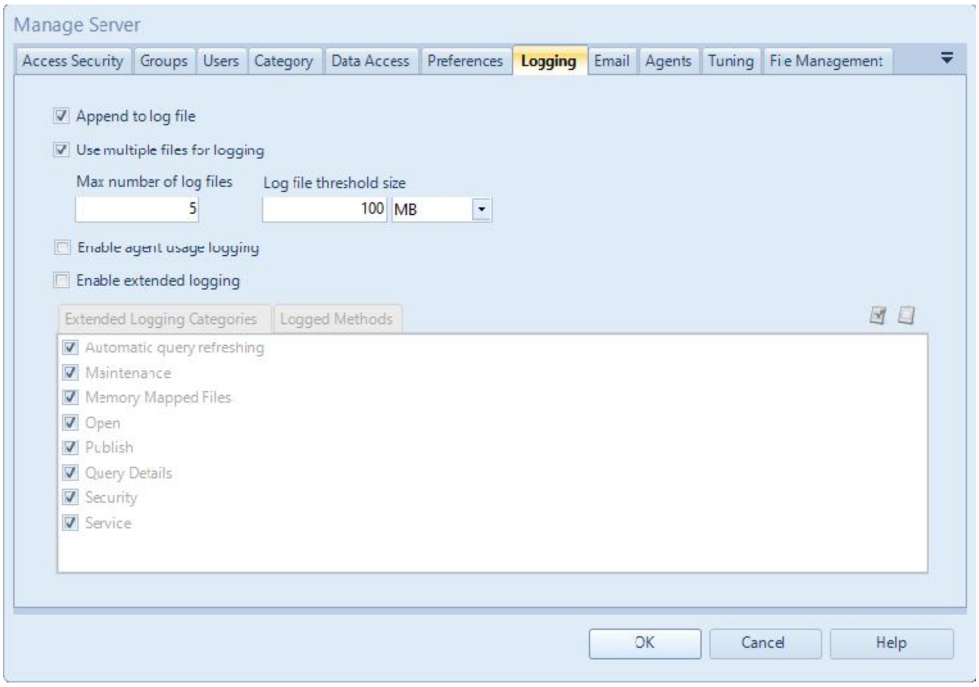
Web page: http://myorganisation.com/

You can also specify some additional text that will be added to the header of every page in DataPA Enterprise. This is useful if you want to add tracking or meta tags to your implementation of DataPA Enterprise. For example, if you wanted to add Google Analytics tracking to DataPA Enterprise, you could do so by adding the following script (substituting UA-XXXXXXX-X with your tracking ID) to the box labelled "Additional text to add to the header section of pages in DataPA Enterprise";

```
<script type="text/javascript">

 var _gaq = _gaq || [];
 _gaq.push(['_setAccount', 'UA-XXXXXXX-X']);
 _gaq.push(['_trackPageview']);

 (function() {
   var ga = document.createElement('script'); ga.type = 'text/javascript'; ga.async =
true;
   ga.src = ('https:' == document.location.protocol ? 'https://ssl' : 'http://www') +
'.google-analytics.com/ga.js';
   var s = document.getElementsByTagName('script')[0]; s.parentNode.insertBefore(ga,
s);
 })();
</script>
```

Logging

The "Logging" tab of the manage server window allows you to configure the level of logging that takes place within the DataPA Enterprise Service.



☑ Append to log file

☑ Use multiple files for logging

Max number of log files: 5

Log file threshold size: 100 MB

□ Enable agent usage logging

□ Enable extended logging

☑ Automatic query refreshing
☑ Maintenance
☑ Memory Mapped Files
☑ Open
☑ Publish
☑ Query Details
☑ Security
☑ Service

The DataPA Enterprise service writes to one or many log files which will be stored in the DataPA folder in one of the following locations, depending on your version of windows:

- C:\ProgramData (newer versions of windows)
- C:\Documents and Settings\All Users\Application Data (older versions of windows)

Using the Append to log file check box you can choose whether you want the log file to be appended to each time the DataPA Enterprise Service starts. If you have this checked then a new log file will be created and overwrite the previous one that has the same file name.

The main log file will be called DataPAEnterpriseService.log or DataPAEnterpriseService.XXXXXX.log where XXXXXX is a sequential number. The number will only be used if you have chosen to break up the log files into a number of separate ones.

To do this you can set the property Max number of log files. This will mean that there will never more log files than the value you specify here. As such it can be used to in conjunction with the Log file threshold size prevent the log files from taking up too much disk space on the machine.

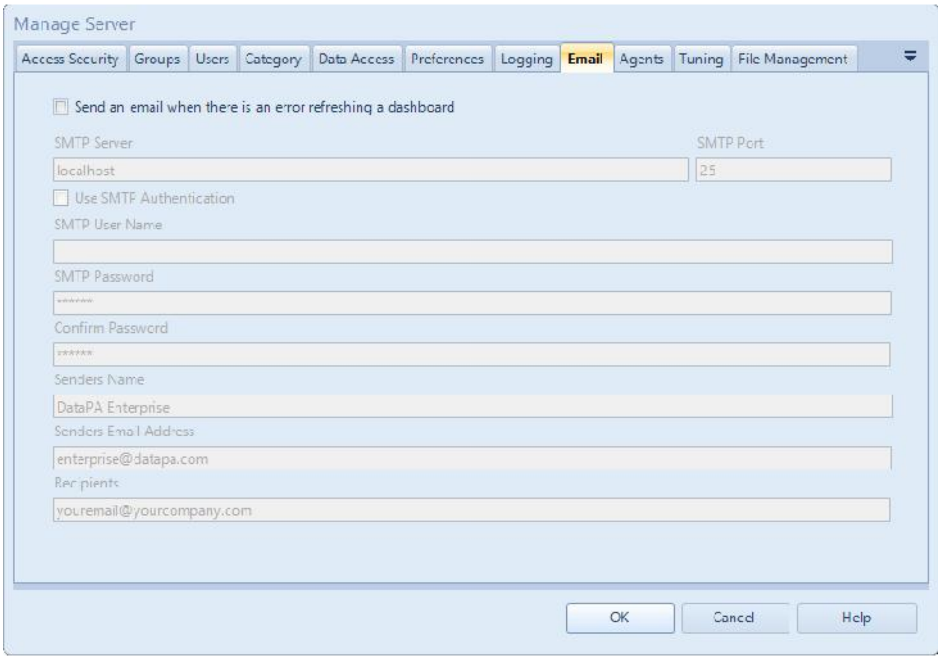
The Log file threshold size allows you to specify the size after which a new log file should be created. This new log file will use the next sequential number in it's file name. If creating a new log file breaching the maximum number set using the Max number of log files value then the oldest log file will be deleted at that point.

The purpose of the logging is mainly to assist in support investigations and the Enable extended logging should normally be switched OFF to improve performance and reduce unnecessary disk usage. In this instance only the stopping and starting of the service and any error messages will be written to the log file and it's growth will be minimal.

Setting the Enable agent usage logging option, writes a line to the log file anytime a request is processed by an agent. This generates data which can then be used to monitor how many agents are in use at any given time.

Email

The server may send emails on your behalf for a number of reasons when Dashboards published to the DataPA Enterprise Service may be *scheduled to refresh automatically*. Firstly, you wish an email to be sent whenever such an automated refresh fails for any reason, then this may be configured in the Email tab of the Manage Server window. Additionally, you may define alerts on dashboards that send emails. The alerts will also use the configuration on this tab to send emails.



□ Send an email when there is an error refreshing a dashboard

SMTP Server: localhost

SMTP Port: 25

□ Use SMTP Authentication

SMTP User Name:

SMTP Password: ******

Confirm Password: ******

Senders Name: DataPA Enterprise

Senders Email Address: enterprise@datapa.com

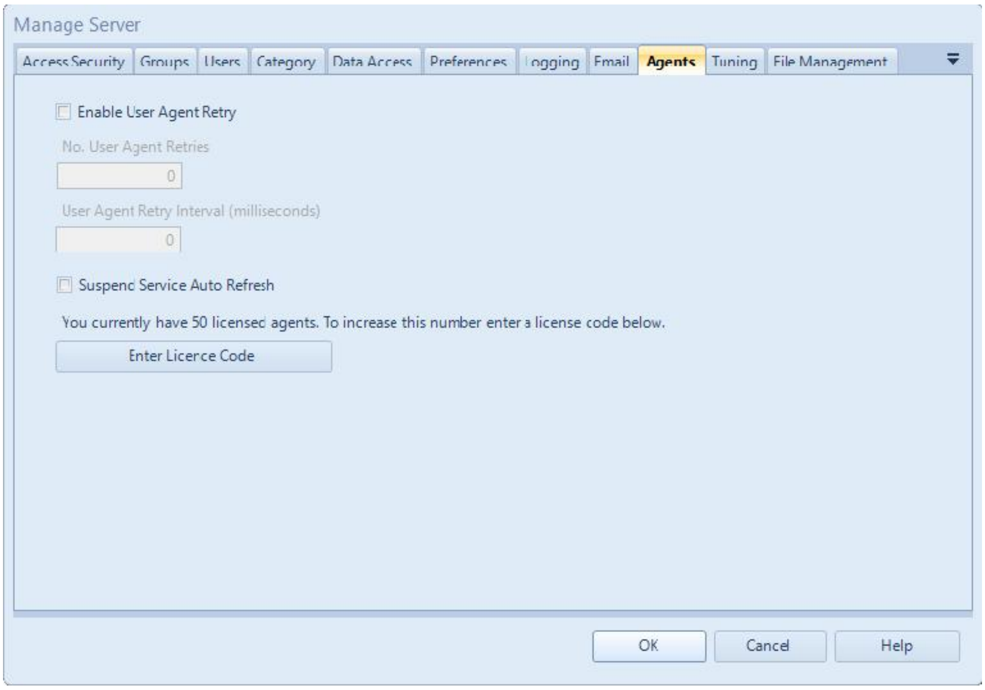
Recipients: youremail@yourcompany.com

The properties to be configured to enable this functionality are as follows:

Property	Description
SMTP Server	The SMTP Server field allows you to enter the network name or IP address of a server on your network that hosts a SMTP Server. This SMTP Server will be used by the DataPA Enterprise Service to send emails.
SMTP Post	The port used by the DataPA Enterprise Service to communicate with the SMTP Server.
Use SMTP Authentication	This checkbox indicates that SMTP Authentication is required to send mail. If selected the SMTP User Name and Password fields become enabled and required.
SMTP User Name	The SMTP user name field allows you to enter the name required for SMTP authentication.
SMTP Password / Confirm Password	The SMTP password / Confirm Password fields allow you to enter and confirm the password required for SMTP authentication.
Senders Name	The senders name field allows you to enter the name that will appear as the senders name in the email.
Senders Email Address	The senders email address field allows you to enter the email address that will appear as the senders name in any emails sent by the DataPA

	Enterprise Service.
Recipients	The email address(es) to whom the emails should be sent. Multiple addresses are separated by semicolons, for example john@domain.com; mary@domain.com

Agents



□ Enable User Agent Retry

No. User Agent Retries: 0

User Agent Retry Interval (milliseconds): 0

□ Suspend Service Auto Refresh

You currently have 50 licensed agents. To increase this number enter a license code below.

To process multiple requests concurrently requires the DataPA Enterprise Service to have multiple agents; as one agent can only process a single request at a time.

A request can be an action such as

- Executing a scheduled automatic refresh of a dashboard or
- Responding to a request for published content by a user.

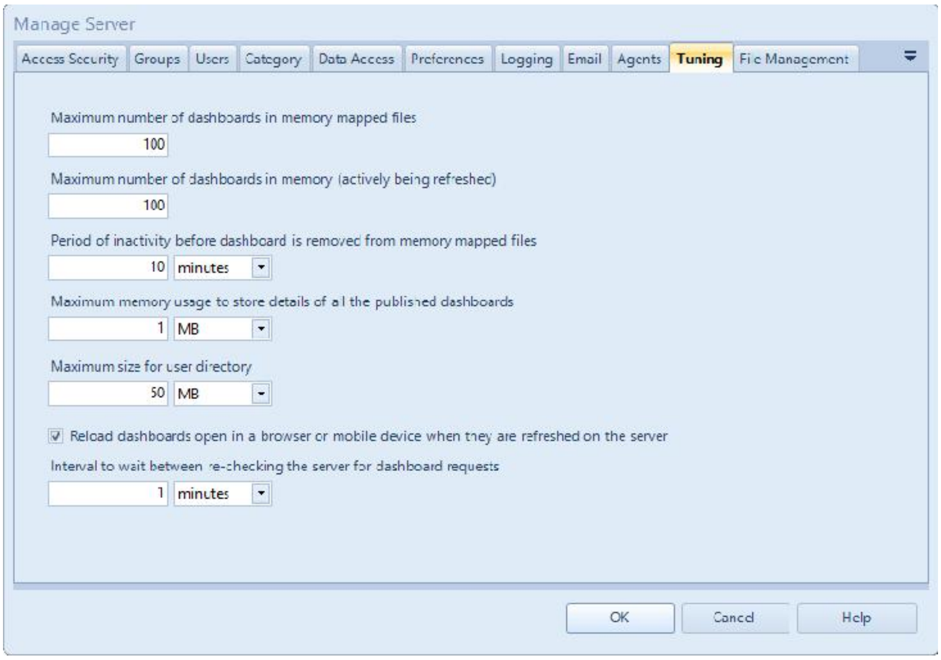
If the DataPA Enterprise Service receives a request, and all agents are currently busy, then the user making the request will receive an error message stating that all agents are busy.

However, if the "Enable User Agent Retry" option is enabled, then DataPA will retry to obtain an agent a set number of times ("No. User Agent Retries"), with a set interval between attempts ("User Agent Retry Interval") before returning the error message if still unable to procure an agent.

The "Suspend Service Auto Refresh", can be used to halt the servicing of all scheduled automatic refreshes. This option can be particularly useful, for instance, if you need to bring your system down for maintenance.

Tuning

The "Tuning" tab of the Manage Server window is intended for use by administrators to control the amount of memory and disk that can be used by the DataPA Enterprise Service.



Maximum number of dashboards in memory mapped files: 100

Maximum number of dashboards in memory (actively being refreshed): 100

Period of inactivity before dashboard is removed from memory mapped files: 10 minutes

Maximum memory usage to store details of all the published dashboards: 1 MB

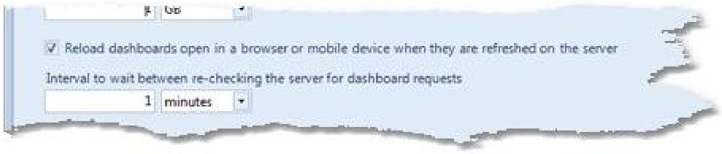
Maximum size for user directory: 50 MB

☑ Reload dashboards open in a browser or mobile device when they are refreshed on the server

Interval to wait between re-checking the server for dashboard requests: 1 minutes

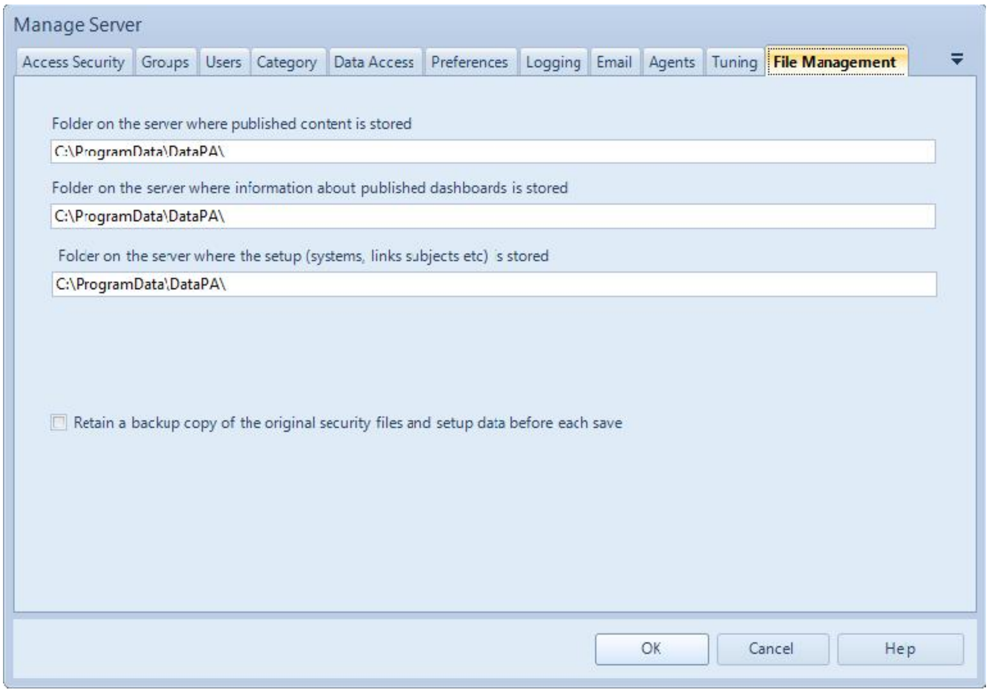
The options available are:

Tuning Option	Description
Maximum number of dashboards in memory mapped files	The DataPA Enterprise Service holds dashboards in memory-mapped files (meaning a portion of memory has been assigned a direct byte-for-byte correlation with the dashboard file) thus improving performance. However this can result in significant memory usage. To control this, you can set the maximum number of dashboards that can be stored in this way. If the maximum is exceeded then when a user requests an additional dashboard to be loaded they will be given an error reporting the issue.
Maximum number of dashboards in memory (actively being refreshed)	The DataPA Enterprise Server also holds dashboards in memory so they can be automatically refreshed by the Server if any dashboards have been published with auto refresh settings. This will also use memory on the machine on which the Server is running. To control this you can set the maximum number of dashboards that can be loaded into memory this way. If the limit is reached then the request to load the additional dashboard will fail and an error will be generated in the log file.
Period of inactivity before dashboard is removed from memory mapped files	This setting allows you to specify the period of time after a Dashboard is last used before unloading it from a it's memory mapped file. This ensures that dashboards are not retained in memory mapped files if they are not being used. The period can be set in seconds minutes or hours. Increasing the period will potentially increase memory usage and performance.
Maximum memory usage to store details of all the published dashboards	The details of all the published content is stored on the server in a memory mapped file providing quick access from both DataPA Enterprise and the DataPA Enterprise Server. A maximum size needs to be provided for all memory mapped files and this setting determines that maximum for this content. If a large amount of content has been

	published then it is possible that the default value of this setting may need to be increased. An error will be received by a user if they attempt to publish content that results in the file exceeding this limit.
Maximum size for user directory	When users refresh queries or reports using DataPA Enterprise, the results are stored in a directory specific to that user. You can set a maximum size per user to prevent these files exceeding the available space on a disk.
Reload dashboards open in a browser or mobile device when they are refreshed on the server	Before publishing a dashboard, you can opt for it to be refreshed automatically at set intervals by the service. If this checkbox is checked, when a dashboard is refreshed by the service any instance of that dashboard opened by a user on a mobile device or in a browser will be refreshed. If the user has made any changes to the view of the dashboard, for instance by selecting a different filter, they will be prompted for confirmation before the dashboard is refreshed as these changes will be lost.  GB ☑ Reload dashboards open in a browser or mobile device when they are refreshed on the server Interval to wait between re-checking the server for dashboard requests 1 minutes
Interval to wait between re-checking the server for dashboard requests	When a dashboard is loaded in a browser or mobile device, the server first checks to see if the dashboard will be refreshed automatically, and if it will, when the refresh is likely to finish. This time is sent to the client along with the dashboard. If the dashboard is still open in the client at this time, the client will begin polling the server to check if the dashboard has been refreshed. You can change the time interval between each request the client makes to the server using this interval. A larger time interval could mean a dashboard takes longer to be refreshed in the client, but will reduce the number of agents used and the load on the server.

File Management

The "File Management" tab of the Manage Server window is intended for use by administrators to control the amount of memory and disk that is used by the DataPA Enterprise Service.



Folder on the server where published content is stored: C:\ProgramData\DataPA\

Folder on the server where information about published dashboards is stored: C:\ProgramData\DataPA\

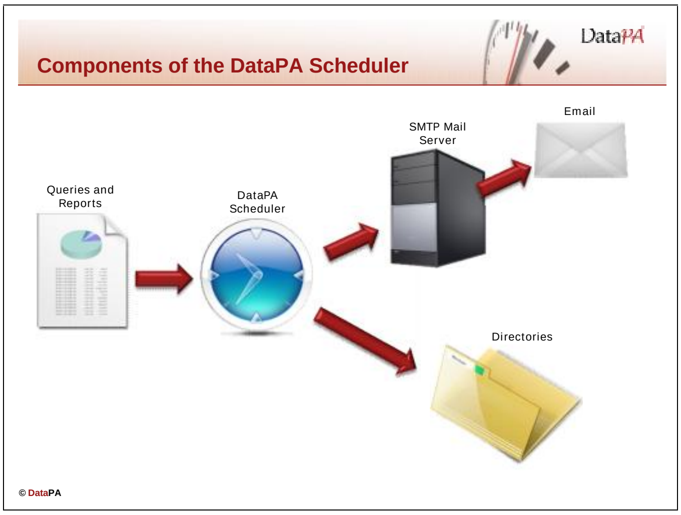
Folcer on the sever where the setup (systems, links sJbjects etc) s stored: C:\ProgramData\DataPA\

□ Retain a backup copy of the original security files and setup data before each save

It is possible to set the following options:

File Management Option	Description
Folder on the server where published content is stored	These are the files holding • Dashboards from the Enterprise Dashboard (*.edp) • Reports from the Report Designer (*.rpa) • Queries from the MS Excel / Access add-ins (*.qpa).
Folder on the server where information about published dashboards is stored	Information on the published dashboards such as last refresh time and status is held in DataPAEnterpriseInfoStore.dat.
Folder on the server where the setup (systems, links, subjects etc.) is stored	These files (Subjects.dat SDOs.dat Partitions.dat Lookups.dat Links.dat Conditions.dat) hold the configuration from the *Modify SetUp* window. This window is used to maintain details of *Systems*, *Links* and *Subjects* for DataPA.
Retain a backup copy of the original security files and setup data before each save	The security settings relating to the server are stored on the server in the DataPA folder located in the Windows common application data folder. The settings are stored in the file named DataPAEnterpriseSecuritySettings.dat. When changes are made to the DataPA Enterprise security setup, before the new configuration is written to the DataPAEnterpriseSecuritySettings.dat file the previous version is renamed to something of the form DataPAEnterpriseSecuritySettings.dat.210201216726691.

	This allows you to restore your setup should you make a mistake. To restore a previous setup, proceed as follows: • Stop the DataPA Enterprise service (Control panel, Services) • Rename DataPAEnterpriseSecuritySettings.dat to some unique nameRename the previous version you wish to restore to DataPAEnterpriseSecuritySettings.dat • Restart the DataPA Enterprise service (Control panel, Services)



Components of the DataPA Scheduler

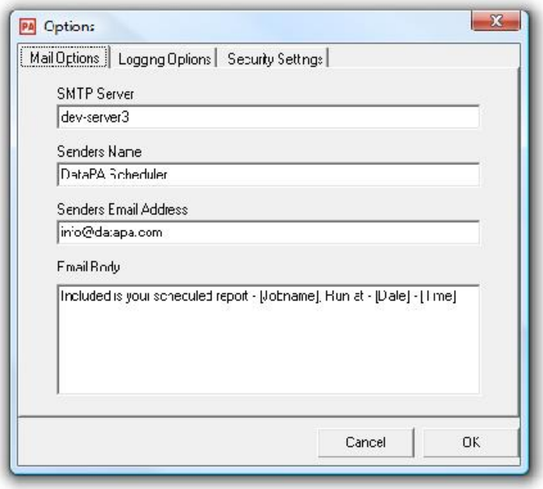
Introduction

The DataPA Scheduler allows users to schedule reports and queries to be run at a particular point in time, and the results either emailed to a list of email addresses, or copied to a specific directory. The scheduler uses the Microsoft Task scheduler to run jobs at a particular point in time, removing the need for any additional services, and will connect to any SMTP mail server to send emails.

Configuring the Scheduler Mail Settings

In order for the DataPA Scheduler to send emails, it needs to be configured to connect to an SMTP server. To configure the Scheduler Mail Settings, follow the steps below:

1. Open the DataPA Scheduler.
2. From the menu, select *Tools →Options.*
3. Select the *Mail Options* tab.
4. Enter the host name of the machine hosting the SMTP service.
5. Enter a from address for emails to be sent from.
6. Enter some text for the email body. (NB> you can use the identifiers [Jobname], [Date], [Time] which will be substituted with the name of the job and the date and time it was run respectively.
7. Press *OK*



SMTP Server: dev-server3

Senders Name: DataPA Scheduler

Senders Email Address: info@datapa.com

Email Body: Included is your scheduled report - [Jobname]. Run at - [Date] - [Time]

Configuring the logging settings

The logging settings allow you to control the location of the log file the Scheduler writes to, and specify whether it should be emailed to a specific address when each job is completed. To configure the Scheduler Logging Settings, follow the steps below:

1. Open the DataPA Scheduler.
2. From the menu, select Tools→Options.
3. Select the *Logging Options* tab.
4. Enter or browse to the name and location of the log file you want the scheduler to use.
5. If you want the log file emailed after each job, check the *Email log file upon completion* checkbox, and enter an email address.
6. Press *OK*

Security Settings

If the scheduler is going to connect to systems that have connection security, you can enter a username and password that the scheduler should use to connect to these systems. To enter a username and password, follow these steps:

1. Open the DataPA Scheduler.
2. From the menu, select Tools→Options.
3. Select the *Security Settings* tab.
4. Enter a username
5. Enter a password
6. Confirm your password.
7. Press *OK*

NB. This username and password is the default username and password used to run a report and query. You can set a specific username and password for each report, using the *Advanced* tab of the *Report Details* dialog box.

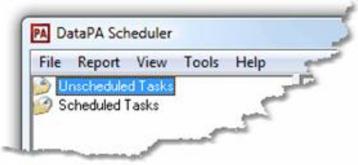
Scheduling Reports and Queries

The DataPA Scheduler allows you to create tasks, which will run one or several reports or queries, and schedule those tasks to be run at particular times. To schedule reports and queries, you need to complete the following steps (each of which will be described in more detail later).

1. Create a new task.
2. Add reports and queries to the task.
3. Schedule a task.

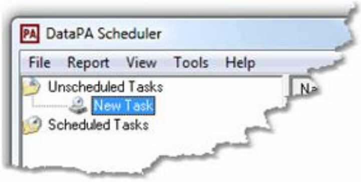
Follow the steps below to schedule a new job:

1. Open the DataPA Scheduler.
2. Select the *Unscheduled Tasks* node in the panel on the left hand side.



3. From the menu, select *File→Add New Task*

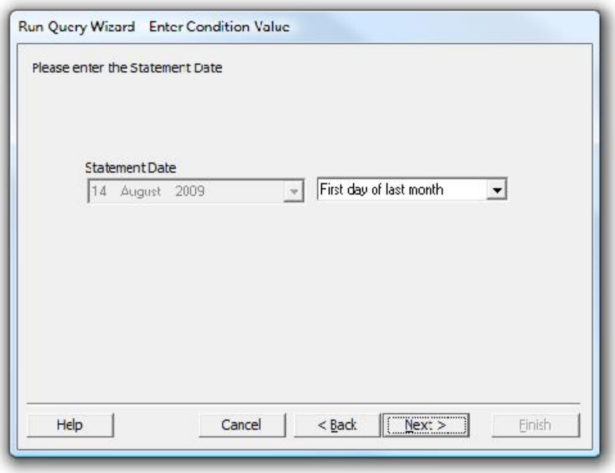
4. Select the new task you created in the panel on the left hand side.



5. Enter an name for your task and press *Enter*.

To add reports and queries to a task, follow these steps:

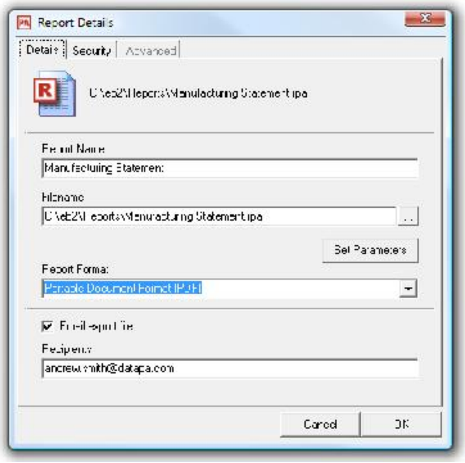
1. Open the DataPA Scheduler.
2. Select your task in the panel on the left hand side.
3. From the menu select *Report →Add Report*
4. Select the browse button (...) to the right of the *Filename* text box, and select the report or query you wish to run.
5. If required, change the name of the report or query (this will only appear within the scheduler).
6. If the report or query has input parameters, press the *Set Parameters* button and complete the query wizard entering the required parameter values.


Run Query Wizard Enter Condition Value

Please enter the Statement Date

Statement Date: 14 August 2009 | First day of last month

7. Select the type of file you want the scheduler to produce from your query or report from the *Report Format* selection list.
8. If you want to email your report, select the *Email export file* option, and enter a list of email addresses (comer separated), or if you want the results to be saved to a file, unselect the *Email export file* option, and browse to the destination file name.
9. Press *OK*


Report Details

C:\eb2\Reports\Manufacturing Statement.rpa

Report Name: Manufacturing Statement

Filename: C:\eb2\Reports\Manufacturing Statement.rpa

Report Format: Portable Document Format (PDF)

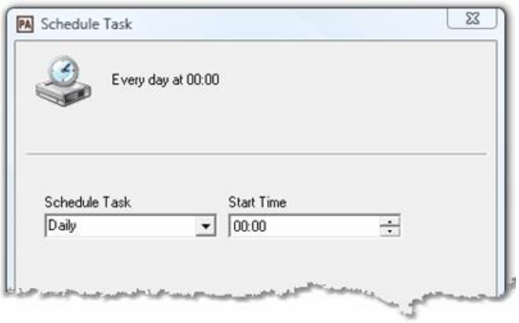
☑ Email export file

Recipients: andrew.smith@datapa.com

You can add as many reports to a single task as you like, by simply repeating the steps above.

To schedule a task, follow these steps:

1. Open the DataPA Scheduler.
2. Select your task in the panel on the left hand side.
3. From the menu select File→Schedule this task.
4. If you want the task run daily, select daily from the drop down list, and select a start time.



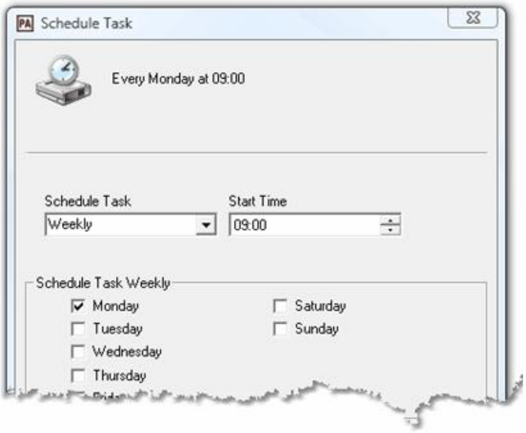
Schedule Task

Every day at 00:00

Schedule Task: Daily

Start Time: 00:00

5. If you want the task run weekly, select weekly from the drop down list, select the days you want the report to be run, and enter a start time.



Schedule Task

Every Monday at 09:00

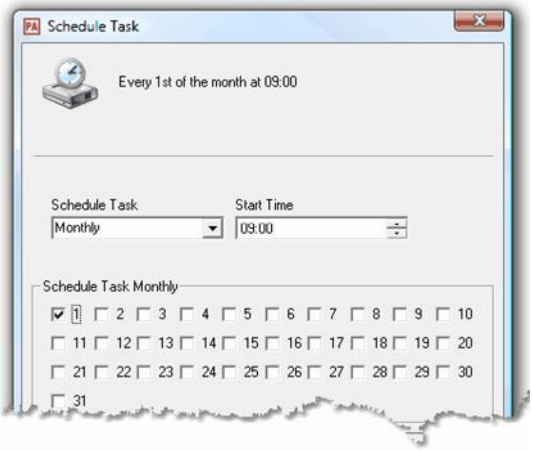
Schedule Task: Weekly

Start Time: 09:00

Schedule Task Weekly

☑ Monday
☐ Tuesday
☐ Wednesday
☐ Thursday
☐ Saturday
☐ Sunday

6. If you want the task run monthly, select monthly from the drop down list, select the days you want the report to be run, and enter a start time.



Schedule Task

Every 1st of the month at 09:00

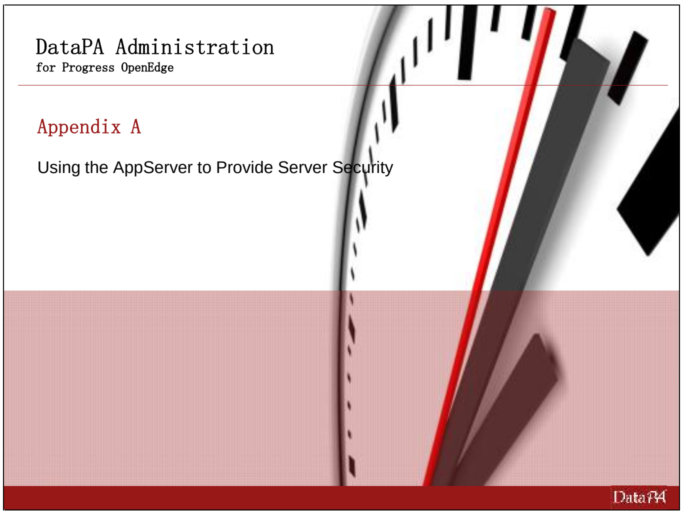
Schedule Task: Monthly

Start Time: 09:00

Schedule Task Monthly

☑ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 ☐ 7 ☐ 8 ☐ 9 ☐ 10
☐ 11 ☐ 12 ☐ 13 ☐ 14 ☐ 15 ☐ 16 ☐ 17 ☐ 18 ☐ 19 ☐ 20
☐ 21 ☐ 22 ☐ 23 ☐ 24 ☐ 25 ☐ 26 ☐ 27 ☐ 28 ☐ 29 ☐ 30
☐ 31

7. Press *OK.*



DataPA Administration
for Progress OpenEdge

Appendix A

Using the AppServer to Provide Server Security

Appendix A: Using the AppServer to Provide Server Security

Introduction

Controlling security access is only effective on the client if the administrator can control security on each individual client machine that can gain access to the server. As a result DataPA allows all security settings to be controlled by the server, preventing any unauthorised actions from any client machines regardless of their configuration. This lesson shows how to use the DataPA server administration techniques to provide connection security, control client security settings, provide user specific set up data and lock access to a server.

Learning Objectives

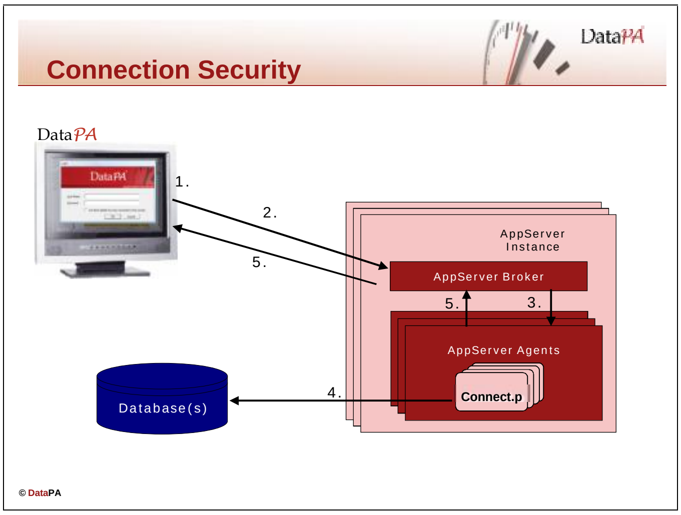
When you complete this lesson, you should be able to:

- Configure and use connection security.
- Configure and use server-side security settings.
- Configure user specific setup data.
- Lock servers so they cannot be accessed by any client with local setup data.

Prerequisites

Before you begin this lesson you should be able to:

- Setup and administer Progress AppServer instances
- Define the DataPA data location as an AppServer connection
- Understand Progress user access security
- Write ABL business logic.



Connection Security

© DataPA

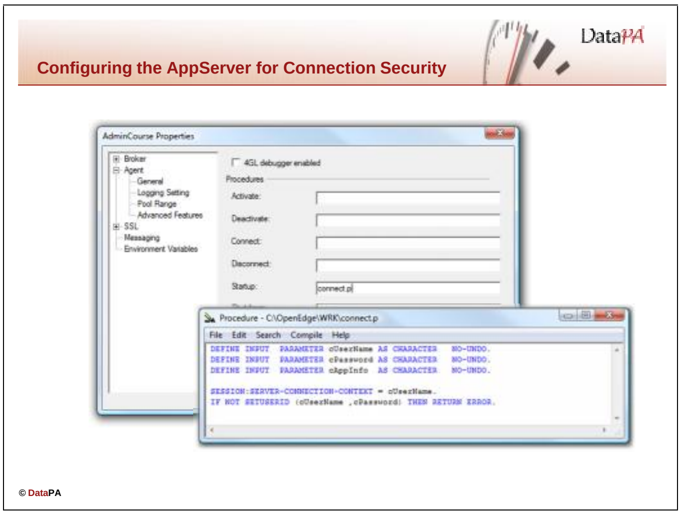
Introduction

Most business applications require some level of access security to ensure users only have access to information that is relevant to their tasks. DataPA can be configured to allow or deny connections to the AppServer based on an existing business applications security, removing the need to maintain user lists in two places.

How Connection Security Works

When using connection security, the following occurs when DataPA attempts to connect to an AppServer instance:

1. When DataPA needs to connect to an AppServer, it first prompts the user for a username and password.
2. DataPA sends a request to the broker to connect to the AppServer. The username and password is included in this request.
3. The broker connects to an agent, and runs a connect procedure, passing it the username and password.
4. The connect procedure accesses the database and determines whether or not the username and password are valid.
5. The connect procedure finishes, returning an error if it did not validate the username and password.
6. If the connect procedure returned an error, the broker returns a connection refused message to DataPA and terminates the connection. Otherwise, the broker returns a connection succeeded message to DataPA and awaits the next request.



Configuring the AppServer for Connection Security

□ 4GL debugger enabled

Procedures

Activate:

Deactivate:

Connect:

Disconnect:

Startup: connect.p

Procedure - C:\OpenEdge\WRK\connect.p

```
DEFINE INPUT  PARAMETER cUserName AS CHARACTER   NO-UNDO.
DEFINE INPUT  PARAMETER cPassword AS CHARACTER   NO-UNDO.
DEFINE INPUT  PARAMETER cAppInfo  AS CHARACTER   NO-UNDO.

SESSION:SERVER-CONNECTION-CONTEXT = cUserName.
IF NOT SETUSERID (cUserName ,cPassword) THEN RETURN ERROR.
```

© DataPA

The connect Procedure

To implement the connection security, you must first create a Progress ABL connect procedure that will validate a username and password. The Connect procedure must receive three string parameters, the username, the password and a third parameter called AppServerInfo to receive any other information.

```
/* ************************** Definitions ************************** */
DEFINE INPUT  PARAMETER cUsername      AS CHARACTER   NO-UNDO.
DEFINE INPUT  PARAMETER cPassword      AS CHARACTER   NO-UNDO.
DEFINE INPUT  PARAMETER cAppServerInfo AS CHARACTER   NO-UNDO.
```

The procedure should then validate the username and password using the standard procdure for the connected business application, and return an error if the username or password is not valid.

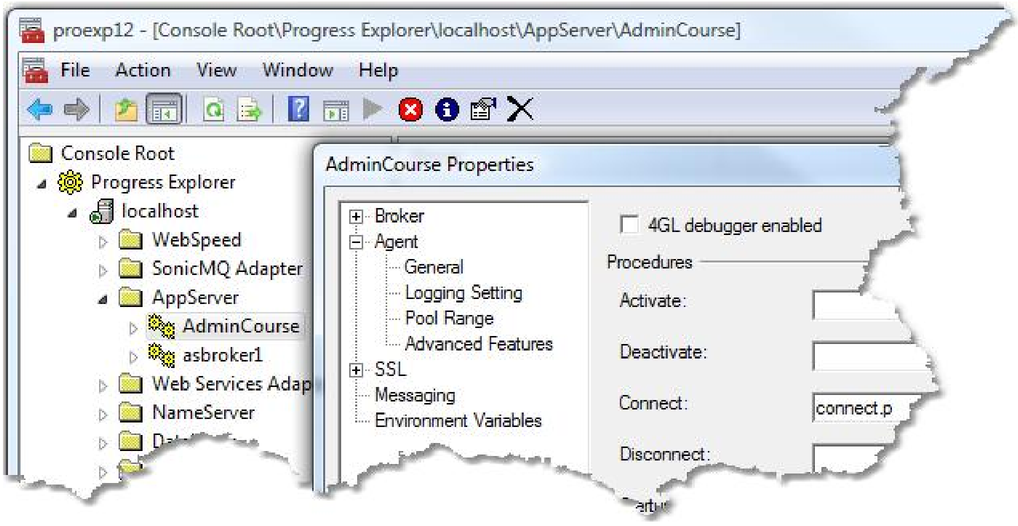
```
/* *********************  Main Block  ********************* */
IF NOT SETUSERID(cUsername,cPassword) THEN RETURN ERROR.
```

Configuring the AppServer

To configure the AppServer to use connection security, follow these steps:

1. Copy the connect procedure detailed above into the PROPATH of the AppServer
2. Start Progress Explorer
    1. Choose the AppServer.
3. Choose *Action* ì *Properties*.
4. Expand the Agent tree from the *Properties* dialog box and select *Advanced Features*

5. Enter the name of the connect procedure in the *Connect* field.



□ 4GL debugger enabled

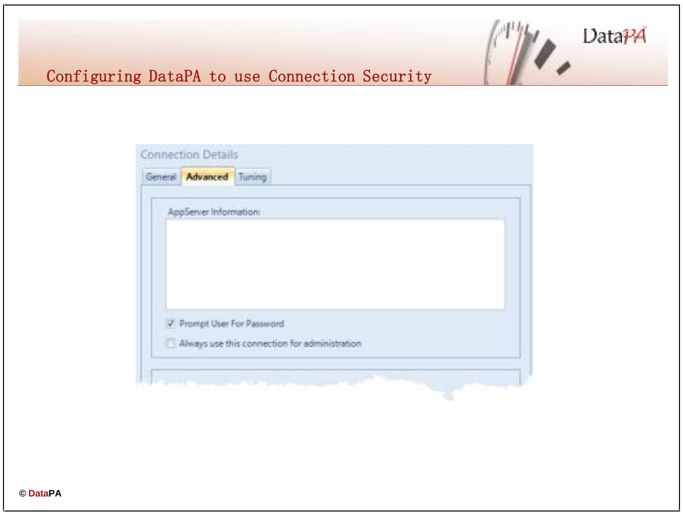
Activate:

Deactivate:

Connect: connect.p

Disconnect:

6. Press OK
7. Restart the AppServer.



Configuring DataPA to use Connection Security

AppServer Information:

☑ Prompt User For Password

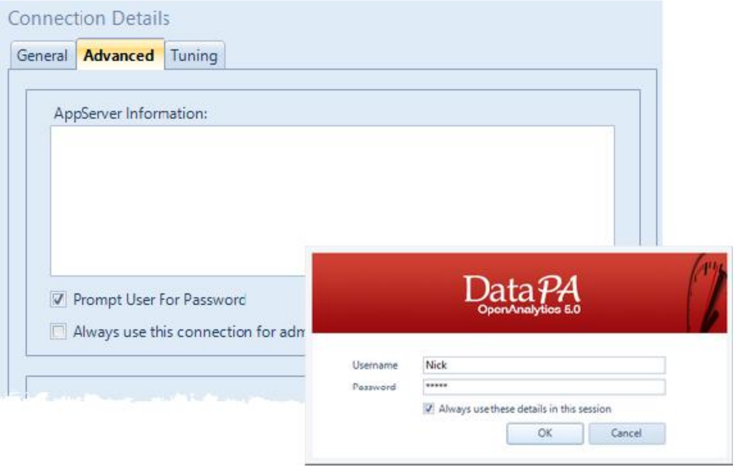
□ Always use this connection for administration

© DataPA

Configuring DataPA to use connection security

Once the AppServer has been configured to process a username and password, the connection must be configured so DataPA prompts the user for a username and password before it tries to connect. Follow these steps to configure a system to prompt for a username and password:

1. Open the DataPA Analytics Engine screen and ensure your Systems tab is selected.
2. Select the system you wish to modify.
3. Select *File → Open*
4. Press *Next* until you reach the *Enter Connection Details* screen
5. Select the default connection, or connection you wish to administer, and press *Edit*
6. Select the *Advanced Tab*
7. Select *Prompt for username and password*
8. Press *OK*
9. Enter a username and password, press *OK*, press *Next*, press *Finish*



AppServer Information:

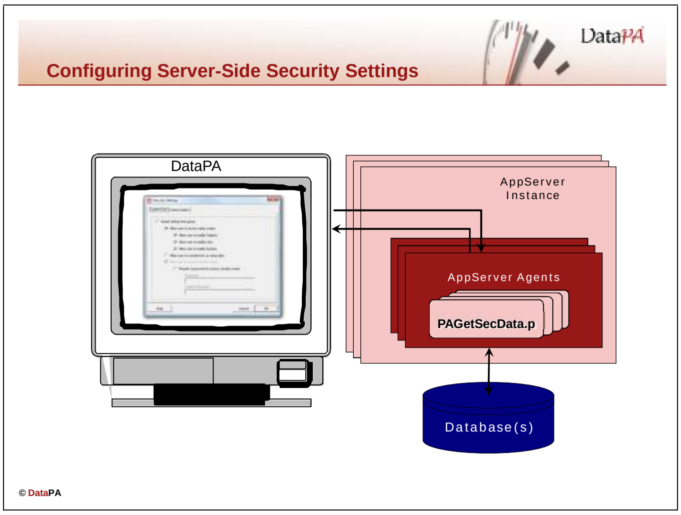
☑ Prompt User For Passworc

□ Always use this connection for adm

Username: Nick

Password: *****

☑ Always use these details in this session



Configuring Server-Side Security Settings

Introduction

Security settings that determine whether or not users can change systems, links and subjects are critical if the security of the business application is to be maintained. To ensure these settings are maintained as designed, regardless of the client machine used to access the system, DataPA allows these settings to be controlled on the server through a business logic procedure.

If DataPA has a data location set to an AppServer (see lesson 3), it will try and retrieve the security settings from the server when it is first executed. To do this it will attempt to run a server-side procedure called PAGetSecData.p. If the procedure does not exist on the server, DataPA will load the security settings from the registry on the client.

The Server Security Procedure

The server security procedure must be called PAGetSecData.p and reside in the PROPATH of the AppServer. The procedure receives the username as an input parameter, and passes back a temp-table with a single record that contains the security settings as an output. The definitions for this procedure should be as follows:

```
DEFINE TEMP-TABLE ttSecData
  FIELD AllowLinks AS INTEGER INITIAL 2
  FIELD AllowSecurity AS INTEGER INITIAL 2
  FIELD AllowSetup AS INTEGER INITIAL 2
  FIELD AllowSubject AS INTEGER INITIAL 2
  FIELD AllowSystem AS INTEGER INITIAL 2
  FIELD RequireSetupPassword AS INTEGER INITIAL 2
  FIELD SetupPassword AS CHARACTER INITIAL ""
  FIELD UserLevel AS CHARACTER INITIAL ""
  FIELD RegUser AS CHARACTER INITIAL ""
  FIELD RegOrganisation AS CHARACTER INITIAL ""
  FIELD RegSerialNum AS CHARACTER INITIAL ""
  FIELD RegCode AS CHARACTER INITIAL ""
  FIELD RegCrystalCode AS CHARACTER INITIAL ""
  FIELD AllowSetupLockOverride AS LOGICAL INITIAL FALSE.

DEFINE INPUT  PARAMETER ip-cUserName     AS CHARACTER  NO-UNDO.
DEFINE INPUT-OUTPUT PARAMETER TABLE FOR ttSecData.
```

The username received by PAGetSecData.p is the username entered by the user when DataPA connects to the AppServer, if available, otherwise the users Windows username. This username can be used to determine the security settings that should be returned.

The Server Security Procedure (Continued ….)

The fields in the ttSecData temp table are as follows:

Field	Description
AllowLinks	Indicates whether or not the user will be able to create, delete and modify links. 0=False,1=True,2=Default
AllowSecurity	Indicates whether or not the user will be able to modify security settings for that session. 0=False,1=True,2=Default
AllowSetup	Indicates whether or not the user will have access to the setup screen. 0=False,1=True,2=Default
AllowSubject	Indicates whether or not the user will be able to create, delete and modify subjects. 0=False,1=True,2=Default
AllowSystem	Indicates whether or not the user will be able to create, delete and modify Systems. 0=False,1=True,2=Default
RequireSetupPassword	Indicates whether or not the user requires a password to access the security screen. 0=False,1=True,2=Default
SetupPassword	The password required to access the security screen if required.
UserLevel	If set to Query, will prevent the user from creating or modifying queries and reports.
RegUser	If the client is not licensed, the user name for the client license to apply.
RegOrganisation	If the client is not licensed, the organization for the client license to apply.
RegSerialNum	If the client is not licensed, the serial number for the client license to apply.
RegCode	If the client is not licensed, the registration code for the client license to apply.
RegCrystalCode	If the client is not licensed, the Crystal Advanced Developer license code to apply to the client.
AllowSetupLockOverride	Set to True to allow the user to override the lock for themselves or another user on setup screen. This will result in the first user being unable to save any changes to the setup files. (Version 3.00.0064 and above only)

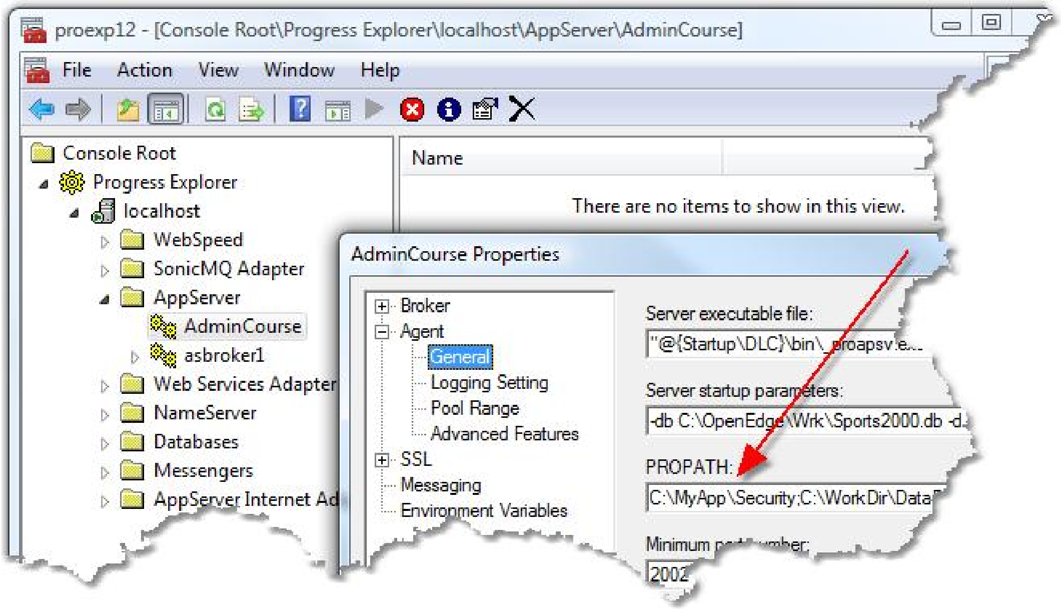
An example of the body code for PASecData.p is as follows:

```
FIND FIRST ttSecData.

AllowSecurity = 0.
AllowSetup = 0.

CASE ip-cUserName:
    WHEN "Nick"   THEN ASSIGN AllowSetup = 1.
    WHEN "Ben"    THEN ASSIGN AllowSetup = 1 AllowSystem = 0.
END CASE.
```

Configuring the AppServer to use Server-Side Security Settings

To configure the AppServer to use Server-Side security settings simply ensure PAGetSecData.p is in the PROPATH for the AppServer. To modify the PROPATH of the AppServer, follow these steps:
Copy the connect procedure detailed above into the PROPATH of the AppServer

1. Start Progress Explorer
2. Choose the AppServer.
3. Choose *Action→Properties*.
4. Expand the Agent tree from the *Properties* dialog box and select *General*
5. Modify the PROPATH



Server executable file: "@{Startup\DLC}\bin_proapsv.e...

Server startup parameters: -db C:\OpenEdge\Wrk\Sports2000.db -d

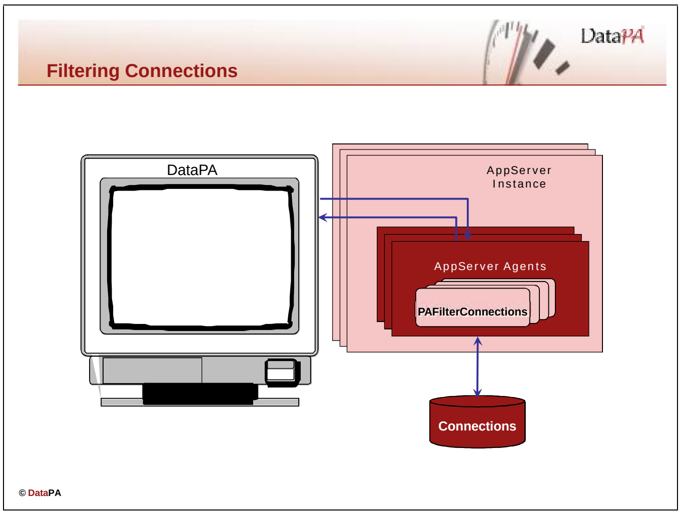
PROPATH: C:\MyApp\Security;C:\WorkDir\Data

Minimum port number: 2002

6. Press OK
7. Restart the AppServer.

Configuring DataPA to use Server-Side Security Settings

If an AppServer is configured correctly to use server-side security, simply set the data location to that AppServer to configure the DataPA client to use server-side security.



Filtering Connections

© DataPA

Introduction

You may want to limit the systems or connections available to particular users, so they can only open and run queries or reports against a certain system, or connection.

The data location determines the location from which the setup files should be loaded. If the data location is set to an AppServer (see lesson 3), DataPA will try to call a procedure called PAFilterConnections.p in the PROPATH of the AppServer, passing the username and a Progress temp-table that contains all the connections. If any of the records are deleted from that temp-table, the corresponding connection will not be available to the user.

The PAFilterConnections Procedure

The PAFilterConnnection procedure must be called PAFilterConnections.p and reside in the PROPATH of the AppServer. The procedure receives the username and the ttConnections temp table an input parameters. The definitions for this procedure should be as follows:

```
DEFINE TEMP-TABLE ttConnections
  FIELD cName           AS CHARACTER
  FIELD cConnectionName AS CHARACTER
  FIELD bPrimary        AS LOGICAL
  FIELD bAdmin          AS LOGICAL.

DEFINE INPUT  PARAMETER ip-cUserName     AS CHARACTER NO-UNDO.
DEFINE INPUT-OUTPUT PARAMETER TABLE FOR ttConnections.
```

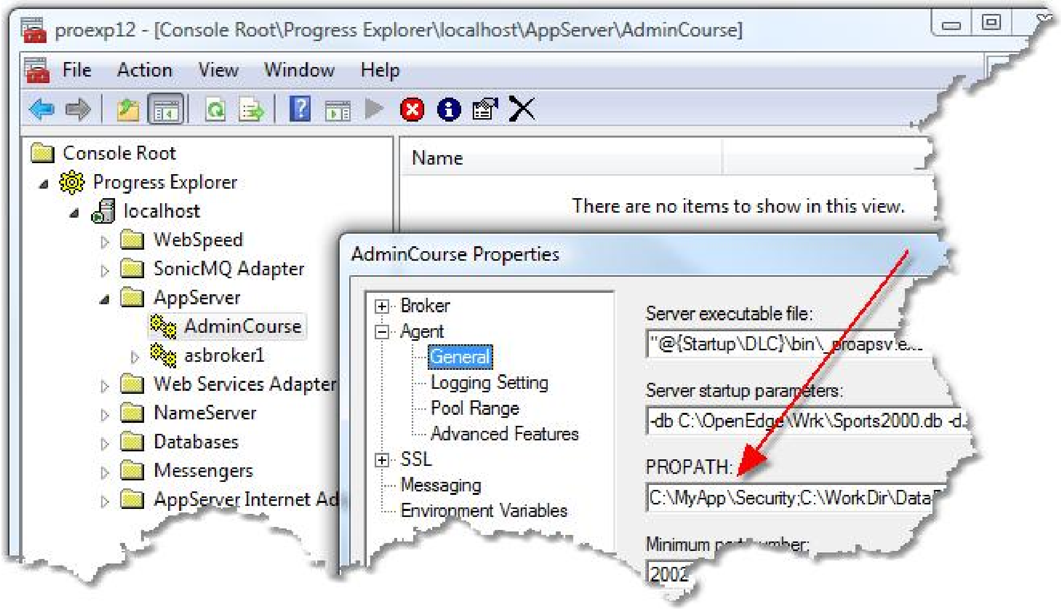
The username received by PAFilterConnnection is the username entered by the user when DataPA connects to the AppServer, if available, otherwise the users Windows username. The procedure can then delete connections you do not want to be available to a user. An example of the body code for PAFilterConnnection.p is as follows:

```
DEFINE VARIABLE cGroups AS CHARACTER   NO-UNDO.

RUN GetUserGroup(INPUT ip-cUserName, OUTPUT cGroups).
FOR EACH ttConnections:
  IF NOT CAN-DO("Finance", cGroups) THEN DO:
    IF cName = "Finance" THEN DELETE ttConnections.
  END.
END.
```

Configuring the AppServer to filter connections

To configure the AppServer to filter connections simply ensure PAFilterConnections.p is in the PROPATH for the AppServer. To modify the PROPATH of the AppServer, follow these steps:

1. Copy the connect procedure detailed above into the PROPATH of the AppServer
2. Start Progress Explorer
3. Choose the AppServer.
4. Choose *Action*→*Properties*.
5. Expand the Agent tree from the *Properties* dialog box and select *General*
6. Modify the PROPATH



Server executable file: "@{Startup\DLC}\bin_proapsv.e...

Server startup parameters: -db C:\OpenEdge\Wrk\Sports2000.db -d

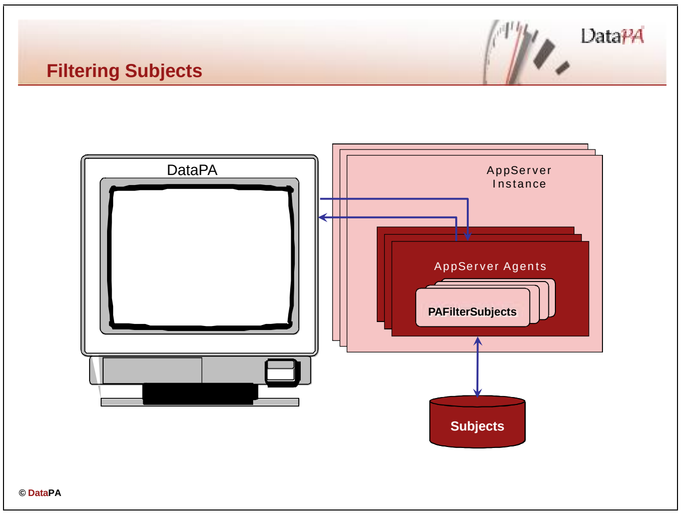
PROPATH: C:\MyApp\Security;C:\WorkDir\Data

Minimum port number: 2002

7. Press OK
8. Restart the AppServer.

Configuring DataPA to use PAFilterConditions

If an AppServer is configured correctly to filter conditions, simply set the data location to that AppServer to configure the DataPA.



Filtering Subjects

DataPA

AppServer Instance

AppServer Agents

PAFilterSubjects

Subjects

© DataPA

Introduction

Users can only create or run queries for which they have subjects available. As such, user data access can be controlled by limiting the subjects. This means it is common to want different subjects available for different users.

The data location determines the location from which the setup files should be loaded. If the data location is set to an AppServer (see lesson 3), and DataPA is configured to use XML setup Data (see lesson 3), DataPA will try to call a procedure called PAFilterSubjects.p in the PROPATH of the AppServer, passing the username and two Progress temp-table that contain all the subjects and their corresponding fields. If any of the subject records are deleted, the subject will not be available to the user and if any of the fields are deleted, the corresponding subject field will not be available to the end user.

The PAFilterSubjects Procedure

The PAFilterSubjects procedure must be called PAFilterSubjects.p and reside in the PROPATH of the AppServer. The procedure receives the username as an input parameter, a ttSubjects and a ttSubjectFields temp table as input-output parameters, and a character output parameter that determines a working directory. The definitions for this procedure should be as follows:

```
DEFINE TEMP-TABLE ttSubjects NO-UNDO
  FIELD cID               AS CHARACTER
  FIELD cTitle            AS CHARACTER
  FIELD cSystemName       AS CHARACTER
  FIELD cDescription      AS CHARACTER
  FIELD cSmartDataObject  AS CHARACTER
  FIELD lDynSDO           AS LOGICAL
  FIELD cTables           AS CHARACTER
  FIELD cParents          AS CHARACTER
  FIELD cBuffers          AS CHARACTER
  FIELD cIndexes          AS CHARACTER
  FIELD cLinks            AS CHARACTER
  FIELD bDynamic          AS LOGICAL FORMAT "true/false"
  INDEX i1 AS PRIMARY UNIQUE cID.

DEFINE TEMP-TABLE ttSubjectFields NO-UNDO
  FIELD cID             AS CHARACTER
  FIELD cTitle          AS CHARACTER
  FIELD cFieldName      AS CHARACTER
  FIELD cDataType       AS CHARACTER
  FIELD cLabel          AS CHARACTER
```

```
    FIELD cFormat        AS CHARACTER
    FIELD lAllowIndex    AS LOGICAL
    FIELD lAllowSort     AS LOGICAL
    FIELD lAllowContains AS LOGICAL
    FIELD iExtent        AS INTEGER
    FIELD iWidth         AS INTEGER
    FIELD cExpression    AS CHARACTER
    FIELD cSvrFormat     AS CHARACTER
    FIELD cLookup        AS CHARACTER
    INDEX i1 AS PRIMARY UNIQUE cID cFieldName.

  DEFINE INPUT  PARAMETER ipcUserName AS CHARACTER NO-UNDO.
  DEFINE OUTPUT PARAMETER opcUseDir   AS CHARACTER NO-UNDO.
  DEFINE INPUT-OUTPUT PARAMETER TABLE FOR ttSubjects.
  DEFINE INPUT-OUTPUT PARAMETER TABLE FOR ttSubjectFields.
```

The username received by PAFilterSubjects is the username entered by the user when DataPA connects to the AppServer, if available, otherwise the users Windows username.

DataPA will sometimes use temporary xml files to process subject filters. If you would like to specify a particular directory for DataPA to use for these temp files, set the opcUseDir output parameter to this directory.

The ttSubjects temp-table contains all the subjects available to the user, and the ttSubjectFields temp-table contains all the fields for each subject joined by the primary index. If you delete any ttSubjects records, the corresponding subject will not be available to the user. If you delete any ttSubjectFields records, those fields will not be available to the end user if they use the subject.

An example of the body code for PAFilterConnnection.p is as follows:

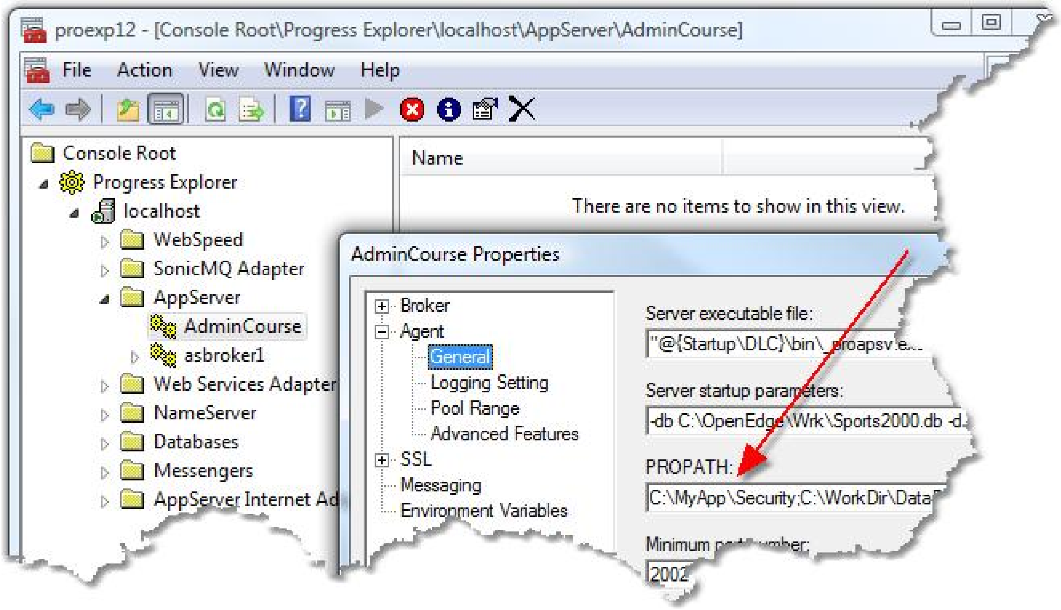
```
  DEFINE VARIABLE lFound AS LOGICAL     NO-UNDO.
  DEFINE VARIABLE cGroups AS CHARACTER   NO-UNDO.

  RUN GetUserGroup(INPUT ipcUserName, OUTPUT cGroups).
  FOR EACH ttSubjects:
    lFound = FALSE.
    FOR EACH ttSubjectFields OF ttSubjects:
      IF NOT CAN-DO("Finance", cGroups)
      AND ttSubjectFields.cFieldName = "sports2000.Customer.Balance"
      THEN lFound = TRUE.
    END.
    IF lFound THEN DELETE ttSubjects.
  END.
```

Configuring the AppServer to filter subjects

To configure the AppServer to filter subjects simply ensure PAFilterSubjects.p is in the PROPATH for the AppServer. To modify the PROPATH of the AppServer, follow these steps:

1. Copy the connect procedure detailed above into the PROPATH of the AppServer
2. Start Progress Explorer
3. Choose the AppServer.
4. Choose *Action→Properties*.
5. Expand the Agent tree from the *Properties* dialog box and select *General*

6. Modify the PROPATH



Server executable file: "@{Startup\DLC}\bin_proapsv.e...

Server startup parameters: -db C:\OpenEdge\Wrk\Sports2000.db -d...

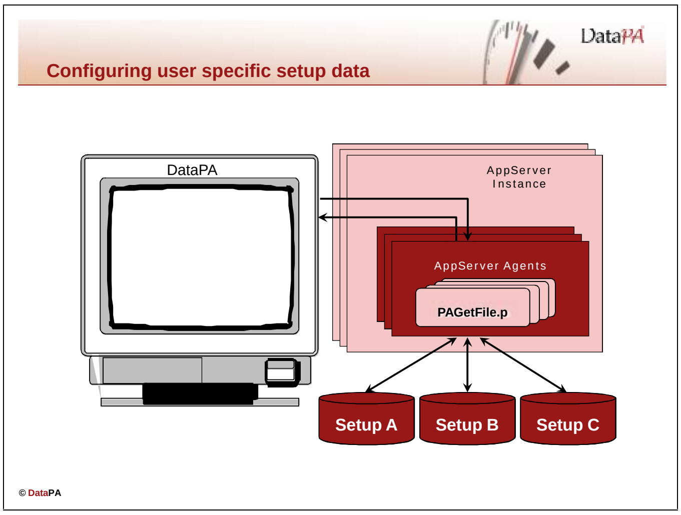
PROPATH: C:\MyApp\Security;C:\WorkDir\Data...

Minimum port number: 2002

7. Press OK
8. Restart the AppServer.

Configuring DataPA to use PAFilterSubjects

If an AppServer is configured correctly to filter conditions, simply set the data location to that AppServer and ensure the data file type is set to *XML*.



Configuring user specific setup data

Introduction

You may want to distribute completely different setup data for different users. The data location determines the location from which the setup files should be loaded. If the data location is set to an AppServer (see lesson 3), DataPA will by default look for the first instance of the setup files in the PROPATH, and send them to the client. This behaviour can be overridden with a server-side procedure that can make decisions on which setup files to send based on the username.

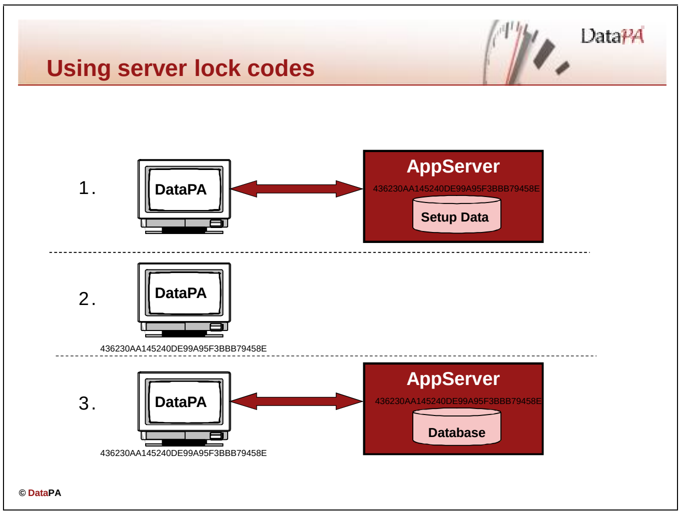
The Server Setup Procedure

The server setup procedure must be called PAGetFile.p and reside in the PROPATH of the AppServer. The procedure receives the username as an input parameter, and a filename as an input-output parameter. The definitions for this procedure should be as follows:

```
DEFINE INPUT        PARAMETER cUsername AS CHARACTER   NO-UNDO.
DEFINE INPUT-OUTPUT PARAMETER cFileName AS CHARACTER   NO-UNDO.
```

The username received by PAGetFile.p is the username entered by the user when DataPA connects to the AppServer, if available, otherwise the users Windows username. PAGetFile.p can then change the filename based on the username. An example of the body code for PASecData.p is as follows:

```
CASE cUsername:
   WHEN "Nick"
     THEN cFileName = "C:\DataPA\Data\Nick\" + cFileName.
   WHEN "Ben"
     THEN cFileName = "C:\DataPA\Data\Ben\" + cFileName.
   WHEN "Freddy"
     THEN cFileName = "C:\DataPA\Data\Freddy\" + cFileName.
   OTHERWISE cFileName = "C:\DataPA\Data\Default\" + cFileName.
END CASE.
```



Using server lock codes

DataPA
1.
DataPA
AppServer
436230AA145240DE99A95F3BBB79458E
Setup Data
2.
DataPA
436230AA145240DE99A95F3BBB79458E
3.
DataPA
AppServer
436230AA145240DE99A95F3BBB79458E
Database
436230AA145240DE99A95F3BBB79458E

© DataPA

Introduction

Applying server-side security is only effective if you ensure any client connecting to the server is subject to that security. Consequently, we need to ensure that a copy of DataPA using its own local security settings and setup data cannot connect to our secure server. To do this we use server lock codes.

How do server lock codes work

Server Lock codes are global unique values set on the AppServer as an environmental variable. If an AppServer has a server lock code defined, it will not allow any DataPA client to connect, unless it has a data location set to an AppServer with an identical lock server code. This means the security settings and setup data were controlled the same, or a matching AppServer.

When DataPA runs a query successfully against an AppServer that is configured with a server lock code, the following steps occur:

1. When DataPA first starts, it connects to an AppServer with a server-lock code to retrieve the setup data.
2. DataPA stores the server-lock code from the AppServer on the DataPA client.
3. DataPA tries to connect to an AppServer to run a query. The AppServer has a server lock code that matches that stored on the client, so DataPA connects ok and runs the query.

When DataPA attempts to run a query using local setup data against an AppServer configured with a server-lock code, the following steps occur:

1. When DataPA first starts, setup data is loaded from the client.
2. DataPA stores a blank server-lock code on the DataPA client.
3. DataPA tries to connect to an AppServer to run a query. The AppServer has a server lock code that does not match the blank code stored on the client, so DataPA fails to connect and cannot run the query.

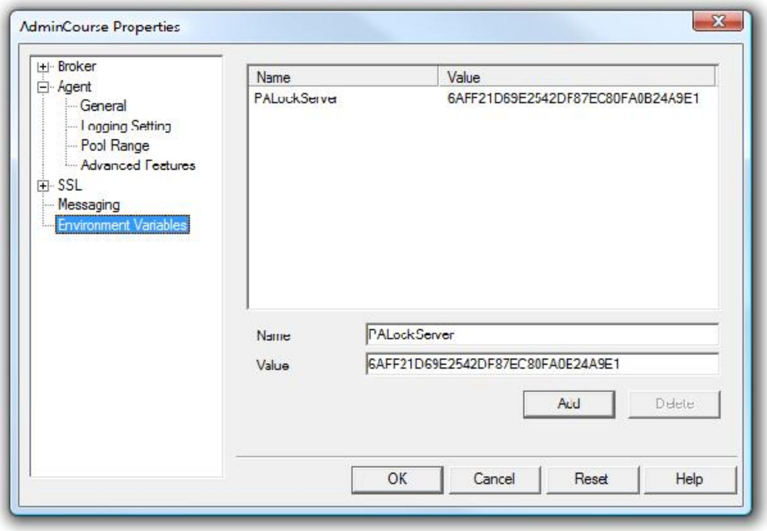
Generating a Server Lock Codes

Like a password, a server lock code can be any character string you like, when creating the code consider that, for the best security, the lock server code should be a string that is impossible to guess and difficult to remember (the end user will never have to remember and type the code).

Applying Server Lock Codes

Follow these steps to apply a server lock code to an AppServer:

1. Start Progress Explorer
2. Choose the AppServer.
3. Choose *Action* ì *Properties*.
4. Select *Environmental Variables*
5. Enter *PALockServer* as the name
6. Paste the lock server code into value and press *Add*

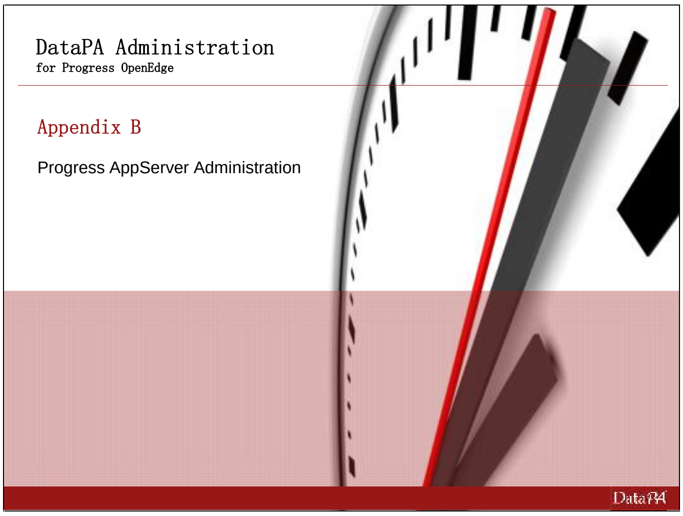

AdminCourse Properties

Name	Value
PALockServer	6AFF21D69E2542DF87EC80FA0B24A9E1

Name: PALockServer

Value: 6AFF21D69E2542DF87EC80FA0E24A9E1

7. Press OK
8. Restart the AppServer



DataPA Administration
for Progress OpenEdge

Appendix B

Progress AppServer Administration

Appendix B: Progress AppServer Administration

Introduction

This lesson shows how to configure and start AppServer components, and how to check their status while they are running. The Progress AppServer allows DataPA to report on complex distributed applications by calling remote procedures from within the DataPA client.

Learning Objectives

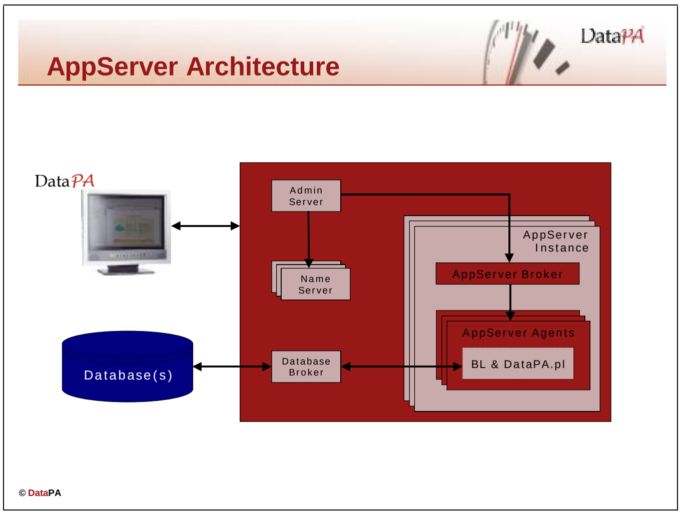
When you complete this lesson, you should be able to:

- Add and configure AppServer components
  - AdminServer
  - NameServer
  - AppServer
- Start AppServer components using the Progress Explorer and check their status. Start the:
  - AdminServer
  - Progress Explorer
  - NameServer
  - AppServer
- Work with the ubroker.properties file and recognize its entries.
- Use command line utilities as an alternative to the Progress Explorer to start, stop, and query the NameServer and AppServer broker.

Prerequisites

Before you begin this lesson you should be able to:

- Administer a Progress database
- Start and stop a Progress Database broker.



AppServer Architecture

Introduction

Before you can connect DataPA to an AppServer, you must be sure that the following components are running on the AppServer host:
AdminServer
Database broker (on the database server host)
NameServer
AppServer
Some or all of these may be set to autostart. Manual start and autostart are described later in this section.

AdminServer

The AdminServer manages the other OpenEdge components. It also allows you to configure and manage components using the Progress Explorer tool or command-line utilities.

NameServer

The NameServer:

- Maintains a list of available AppServers and the application services they support.
- Directs a DataPA connection request to an AppServer based on the requested Application Service Name.
- Balances DataPA workload among AppServers.

Database broker

The database broker manages database servers. This course will not cover database brokers.

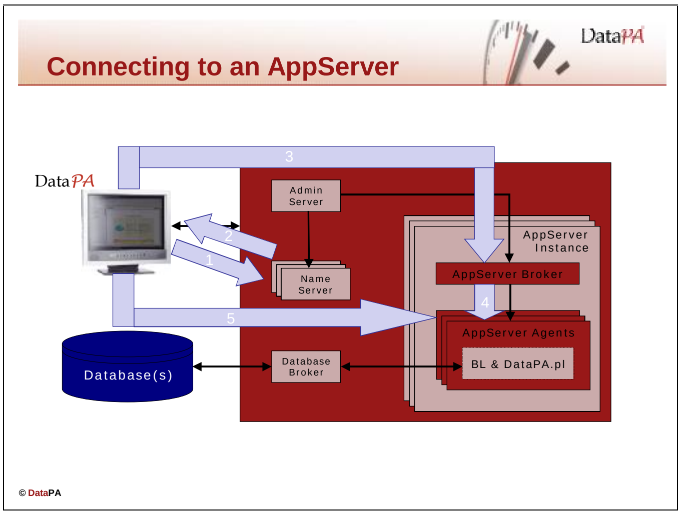
AppServer

The AppServer has two components:

AppServer Broker.

- Manages connections between DataPA clients and a pool of AppServer Agents.
- Maintains a pool of AppServer Agents.
- Routes DataPA client requests to an available AppServer Agent.
- Starts up AppServer Agents as needed to service multiple DataPA clients.
- Trims AppServer Agents when demand is low.

AppServer agents execute DataPA client requests on the server.



Connecting to an AppServer

© DataPA

The connection process

Progress performs several operations using various components when DataPA connects to an AppServer. Generally, the following steps occur. The details may vary somewhat if DataPA is set to connect directly to the AppServer not via the NameServer.

Step	Description
1.	DataPA contacts the NameServer on the AppServer host with the name of an Application Service that DataPA would like to use.
2.	The NameServer returns the information DataPA needs to connect with an AppServer instance that supports the requested Application Service.
3.	DataPA establishes a connection with the AppServer Broker.
4.	The AppServer Broker establishes a connection to an appropriate AppServer Agent.
5.	The broker routes requests to AppServer agents.

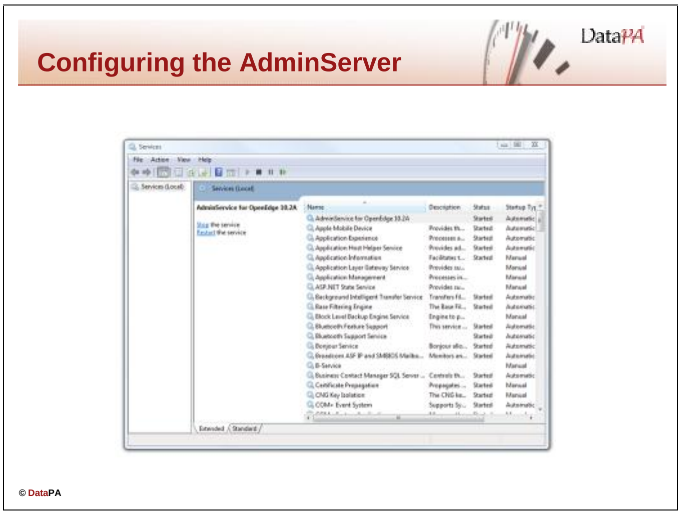
These steps indicate why specific AppServer components must be running before DataPA can connect to the AppServer.

AppServer instances

AppServer instances include an AppServer Broker and a pool of AppServer Agents. The AppServer Agents run on the AppServer host and execute ABL procedures in response to DataPA requests. The AppServer Broker manages the DataPA connections and dispatches requests to AppServer Agents.

Application Service name

An *Application Service Name* is an alias for the AppServer instance that provides a specific business function. For example, an AppServer instance might provide access to the Order Table. You might therefore name the Application Service alias “Orders”. Each instance is defined as a connection within a system in DataPA, and DataPA establishes a connection to the AppServer instance’s AppServer Broker when a query is run against that system. The broker establishes a connection to an AppServer Agent.



Configuring the AdminServer

Services

AdminService for OpenEdge 10.2A

Stop the service
Restart the service

Name	Description	Status	Startup Ty
AdminService for OpenEdge 10.2A		Started	Automatic
Apple Mobile Device	Provides th...	Started	Automatic
Application Experience	Processes a...	Started	Automatic
Application Host Helper Service	Provides ad...	Started	Automatic
Application Information	Facilitates t...	Started	Manual
Application Layer Gateway Service	Provides su...		Manual
Application Management	Processes in...		Manual
ASP.NET State Service	Provides su...		Manual
Background Intelligent Transfer Service	Transfers fil...	Started	Automatic
Base Filtering Engine	The Base Fil...	Started	Automatic
Block Level Backup Engine Service	Engine to p...		Manual
Bluetooth Feature Support	This service ...	Started	Automatic
Bluetooth Support Service		Started	Automatic
Bonjour Service	Bonjour allo...	Started	Automatic
Broadcom ASF IP and SMBIOS Mailbo...	Monitors an...	Started	Automatic
B-Service			Manual
Business Contact Manager SQL Server ...	Controls th...	Started	Automatic
Certificate Propagation	Propagates ...	Started	Manual
CNG Key Isolation	The CNG ke...	Started	Manual
COM+ Event System	Supports Sy...	Started	Automatic

© DataPA

AdminServer on Windows

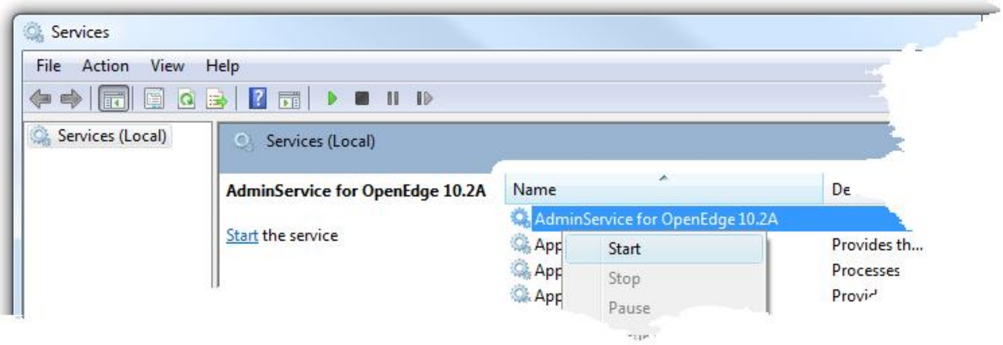
The AdminServer must be running before you start Progress processes on a host machine. The AdminServer defaults to automatically start when the system starts if installed on Windows machines. Follow these steps to see if it is running on your Windows system:

Choose Start ⟶ Settings ⟶ Control Panel ⟶ Administrative Tools ⟶ Services

Manually starting the AdminServer

To start the AdminServer:

1. Select the AdminService for OpenEdge.
2. Choose Action⟶Start.



Services

AdminService for OpenEdge 10.2A

Start the service

Name	De
AdminService for OpenEdge 10.2A	
App	Provides th...
App	Processes
App	Provid

Stopping the AdminServer

To stop the AdminServer on Windows:

1. Select the AdminService for OpenEdge.
2. Choose Action⟶Stop.

UNIX, Linux and Windows (Character or command line mode)

Use the PROADSV utility to start and stop the AdminServer on a UNIX or Linux system or from the command line on a Windows system.

PROADSV has the following syntax:

```
proadsv { { { -start { [ -adminport port-number ] }
| -stop | -query } [ -port port-number ] } | -help }
```

start – Starts the AdminServer.

adminport *port-number*–- Specifies the port number used by the AdminServer for database broker communication. If a port number is not specified, the adminport defaults to port 7832.

stop – Stops the AdminServer.

query – Displays AdminServer status.

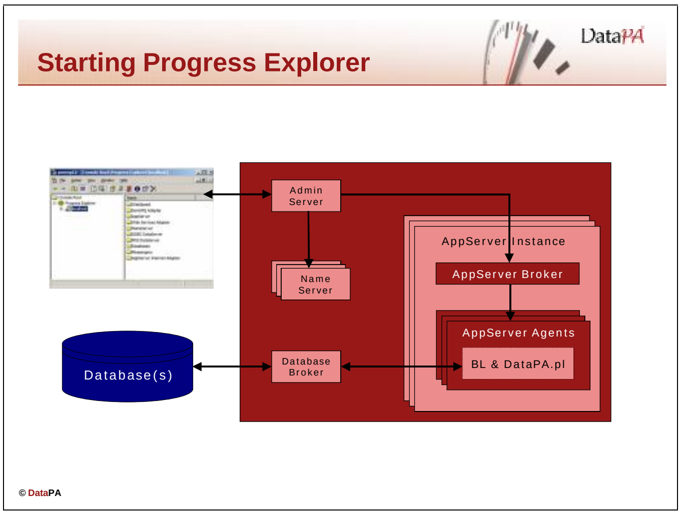
port *port-number* – Specifies the listening port number for online command utilities, such as DBMAN. If a port number is not specified, it defaults to 20931.

To check whether the AdminServer is running on UNIX systems, run the ps command to show the full command line for each process on the system and locate any jre commands in the list. The AdminServer process is running if you see a `jre` command with the arguments that correspond to those specified for `jvmstart` in the Progress `proadsv` shell script located in `OpenEdgeInstallDir/bin`.

PROADSV does not change the windows registry and so the settings you establish are not permanent.

Admserv.log file

The AdminServer writes information to a log file, `admserv.log`, in the Progress working directory (OpenEdge\Wrk by default). The log file keeps a record of AdminServer events and actions, such as starting and stopping AppServer components, and can be a useful troubleshooting aid. You can open `admserv.log` in any text editor.



Starting Progress Explorer

© DataPA

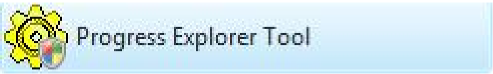
What is the Progress Explorer?

The Progress Explorer is a graphical tool that you use to configure, start, and stop, and retrieve status for NameServers, AppServers, and database connections. The Progress Explorer reads and writes configuration information into files named `conmgr.properties` for databases and `ubroker.properties` for Appservers and Nameservers. Ubroker.properties will be described in more detail later in this lesson. The Progress Explorer works with the AdminServer on the AppServer host to manage AppServers and NameServers.

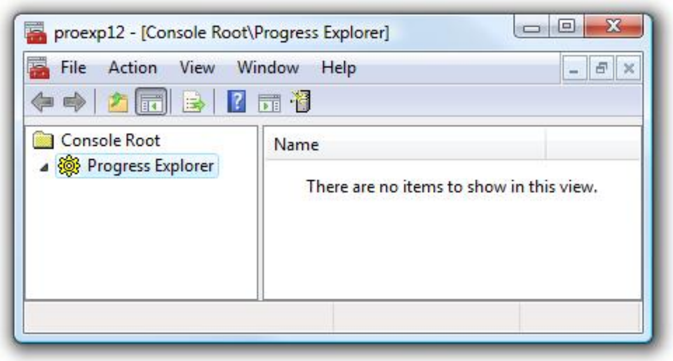
Starting the Progress Explorer

Follow these steps to start the Progress Explorer:

1. Choose Start→Programs→OpenEdge→Progress Explorer tool. (The Progress Explorer Tool is on the OpenEdge menu):



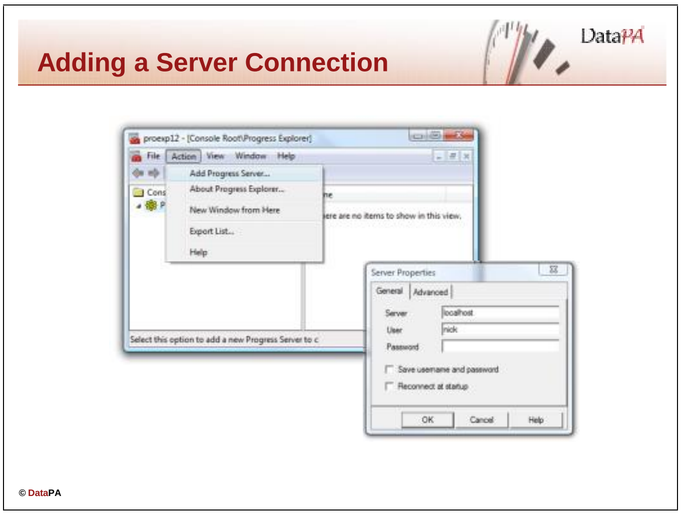
The first time you start the Progress Explorer Tool, it does not have any AppServer connections defined as shown below:



There are no items to show in this view.

Stopping the Progress Explorer

To stop the Progress Explorer:
Choose Console→Exit.



Adding a Server Connection

DataPA
proexp12 - [Console Root\Progress Explorer]
File
Action
View
Window
Help
Add Progress Server...
About Progress Explorer...
New Window from Here
Export List...
Help
Cons
P
ne
ere are no items to show in this view.
Select this option to add a new Progress Server to c
Server Properties
General
Advanced
OK
Cancel
Help

Server: localhost
User: nick
Password:
□ Save username and password
□ Reconnect at startup

© DataPA

The Progress Explorer and the AdminServer

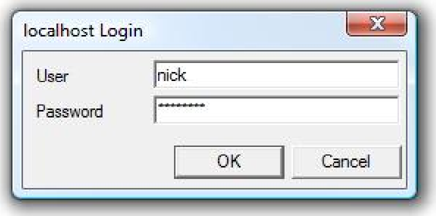
The Progress Explorer tool works with the AdminServer on the AppServer host. Therefore, the Progress Explorer must connect to a running AdminServer before you can use it to manage the AppServer components.

Defining a server connection

To define a new connection to the AppServer host AdminServer:

1. Select Programs→OpenEdge→Progress Explorer Tool from the start menu to start Progress Explorer if it is not already running.
2. Select Action→Add Progress Server from the menu in Progress Explorer.
3. Enter the server host name. (If it is the same as the client system, use localhost.)
4. Enter an appropriate user name and password for the AppServer host system as required.

Connecting to the AdminServer



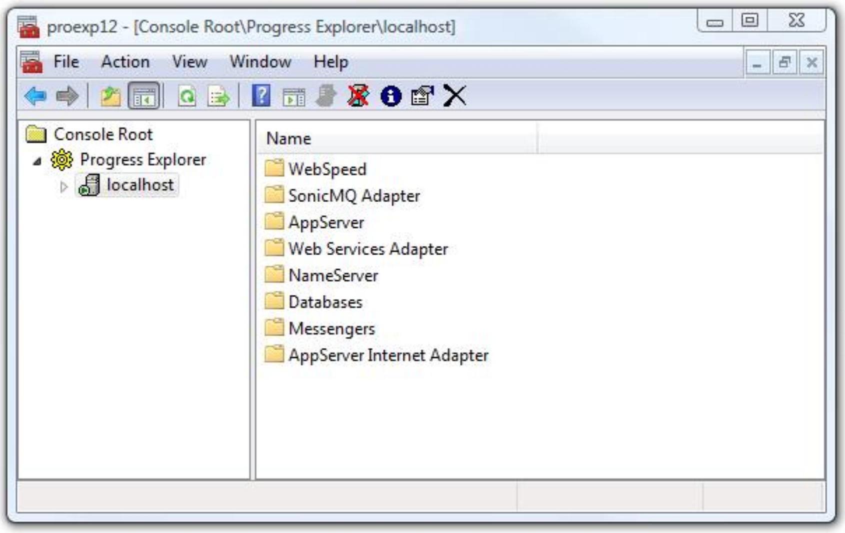
User: nick
Password: *********

To connect to the AdminServer:

1. Select the AdminServer connection icon.
2. Choose Action→Connect.
3. Type your user name and password.
4. Choose OK.

These steps assume the AdminServer connection has been configured as described above.

After connecting, the Progress Explorer window should resemble the following:



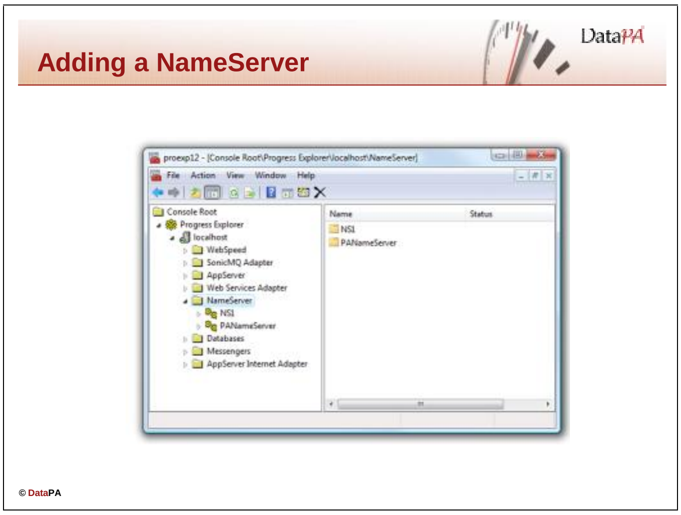
The Progress Explorer window now includes folders for a number of system components, including AppServer components (Database connections, AppServer, NameServer).

Note: If your AdminServer does not use the default port, possibly because you run multiple AdminServers for different versions of OpenEdge or Progress, you can change the AdminServer Connection port number by choosing Action→Properties→Advanced from the Progress Explorer menu.

Disconnecting from the AdminServer

To disconnect from the AdminServer:

1. Select the Server icon.
2. Choose Action→Disconnect.



Adding a NameServer

© DataPA

Introduction

The NameServer directs DataPA client requests to connect to an appropriate AppServer instance. You can have multiple NameServers, providing one level of fault-tolerance and load-balancing. See OpenEdge documentation for further details.

OpenEdge provides one NameServer, NS1, by default. It is recommended that you create your own NameServers, to avoid problems if NS1 changes with OpenEdge updates.

NameServer log files

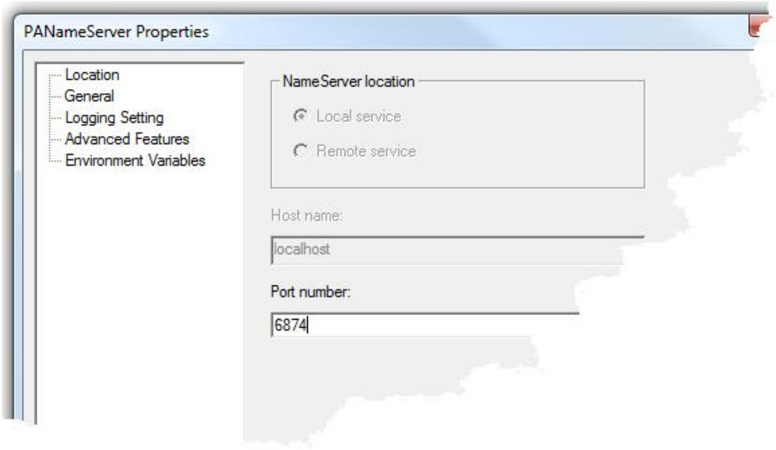
As with the other components, the NameServer creates a log file. During configuration, you can choose where to store the file, how much information to write to the file, and whether to append new information or overwrite old information.

Adding a NameServer with the Progress Explorer

Use the following steps to add a NameServer:

1. Start the Progress Explorer and connect to the AdminServer if needed.
2. Select the NameServer folder.
3. Select Action→New from the Progress Explorer menu.
4. Type the name of the NameServer.
5. Select Location Local.
6. Select OK.
7. Select the new NameServer.
8. Select Action→Properties from the Progress Explorer menu.

9. Enter a Port Number.
   Note: If the Port Number is greyed out, it is already in use. Enter a different number.

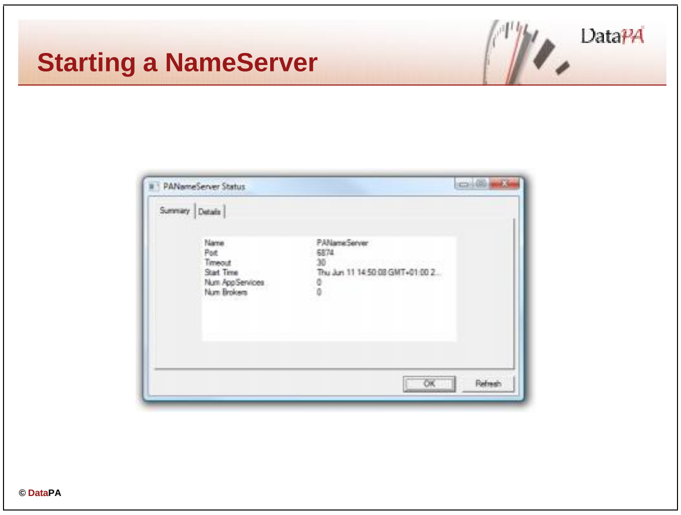


PANameServer Properties

Host name: localhost

Port number: 6874

10. Choose Logging Setting.
11. Enter the server log filename.
12. Disable Append to NameServer log file.
13. Choose OK.
14. Verify that the NameServer displays in the TreeView as shown in the slide.



Starting a NameServer

PANameServer Status

Summary | Details

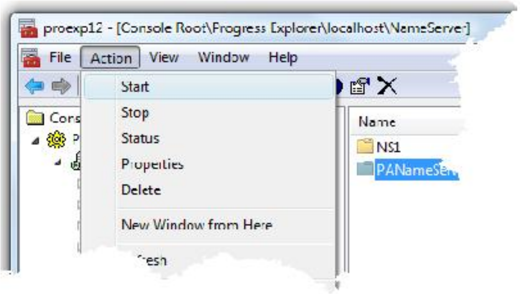
Name PANameServer
Port 6874
Timeout 30
Start Time Thu Jun 11 14:50:08 GMT+01:00 2...
Num AppServices 0
Num Brokers 0

© DataPA

Manually starting a NameServer

Follow these steps to start a NameServer on your system:

1. Select the NameServer icon in the TreeView.
2. Select the name of your NameServer in the Name column.
3. Select Action→Start from the Progress Explorer Menu.
   Note: Allow time for the NameServer to start.
4. Select Action→Status. From the Progress Explorer Menu

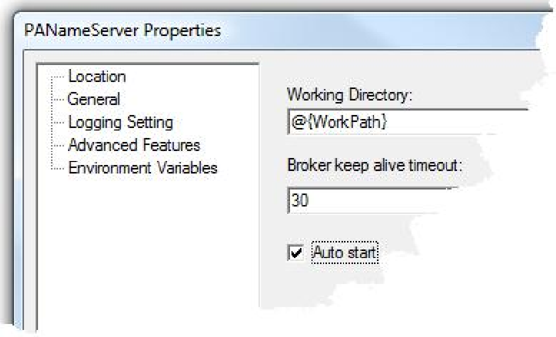


Changing to auto start

When the AdminServer starts, it looks for any other components set to autostart. When all components are set to autostart, the AdminServer starts databases first, then NamerServers, and finally AppServers. Use the following procedure to set the Nameserver to autostart the next time the AdminServer starts:

Follow these steps to change to auto start:

1. Select the NameServer.
2. Select Action→Properties→General from the Progress Explorer Menu.
3. Enable Auto-Start.
4. Choose OK.

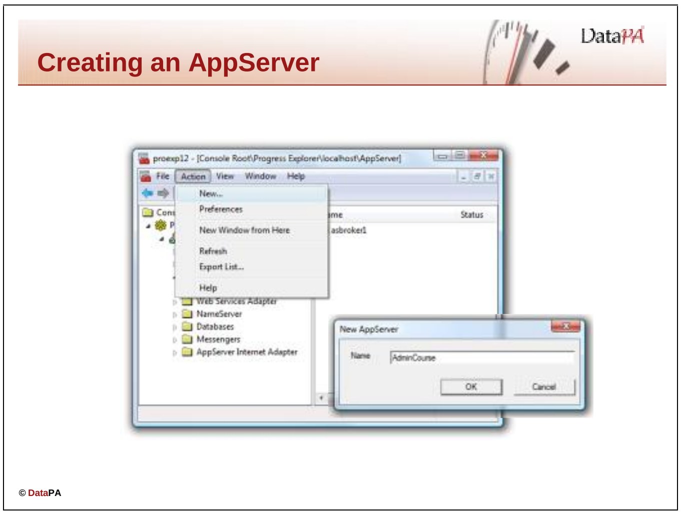


PANameServer Properties

Working Directory: @{WorkPath}

Broker keep alive timeout: 30

☑ Auto start

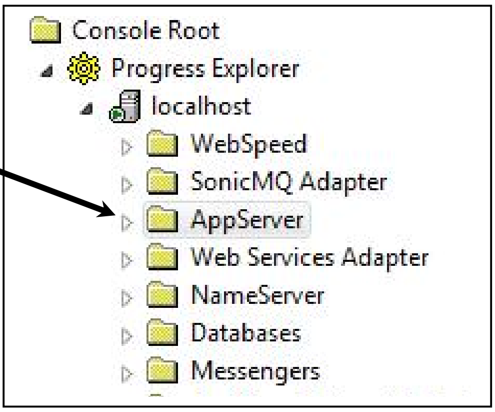


Creating an AppServer

© DataPA

AppServer instances

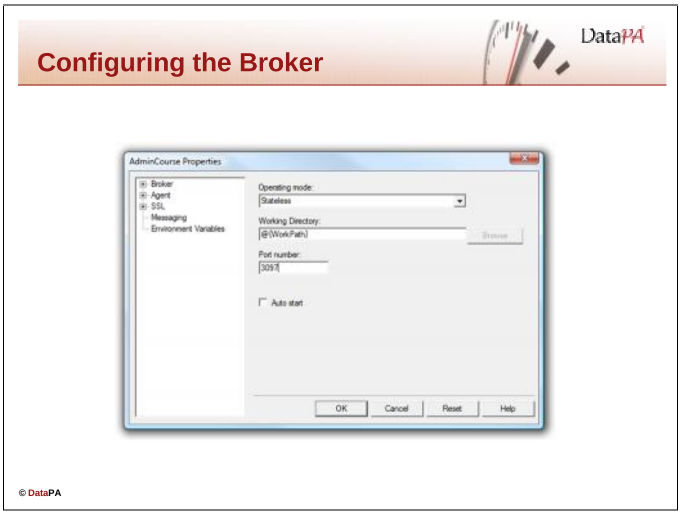
AppServer instances include an AppServer Broker and a pool of AppServer Agents. The AppServer Agents run on the AppServer host and execute ABL procedures in response to DataPA client requests. The AppServer Broker manages the DataPA connections and dispatches requests to AppServer Agents.



Creating an AppServer

To create an AppServer, follow these steps:
Select the AppServer folder in the treeview.

1. Select Action→New from the menu.
2. Enter a name for the AppServer
3. Choose OK.



Configuring the Broker

AdminCourse Properties

Operating mode: Stateless

Working Directory: @{WorkPath}

Port number: 3097

□ Auto start

© DataPA

Configuring the AppServer Broker

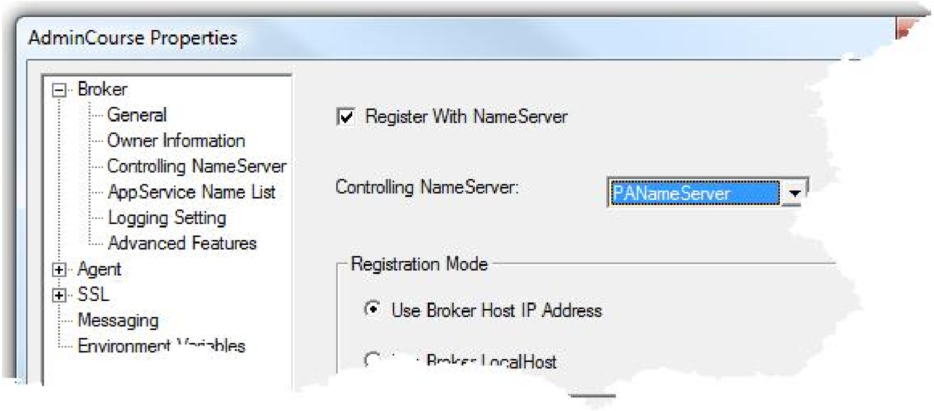
The AppServer Properties dialog has configuration settings for the AppServer Broker and the AppServer Agents.

To configure the AppServer Broker:

1. Select your AppServer in the Progress Explorer tree view.
2. Select Action→Properties from the menu.
   (Or Right Click on the AppServer icon and choose Properties.)
3. Expand the Broker tree view.
4. Choose General to configure the following settings:

Setting	Description
Operating Mode	DataPA recommend Stateless, although all operating modes are supported.
Working Directory	Defaults to the install working directory. Can be changed to the appropriate directory on the AppServer Host.
Port Number	The number of the TCP/IP port that the AppServer Broker listens on. Entry greyed out if port already used or not allowed.
AutoStart	AppServer starts when the AdminServer starts.

5. Choose Controlling NameServer and register the AppServer with the required NameServer. Use Broker Host IP Address.



AdminCourse Properties

☑ Register With NameServer

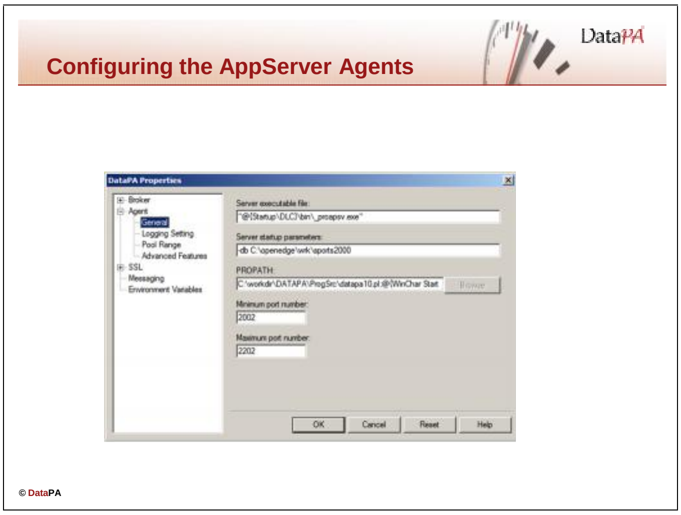
Controlling NameServer: PANameServer

Registration Mode

● Use Broker Host IP Address

○ ··· Broker LocalHost

6. Choose Logging Settings and select the appropriate logging settings.



Configuring the AppServer Agents

DataPA Properties

Server executable file: "@{Startup\DLC}\bin_proapsv.exe"

Server startup parameters: -db C:\openedge\wrk\sports2000

PROPATH: C:\workdir\DATAPA\ProgSrc\datapa10.pl;@{WinChar Start

Minimum port number: 2002

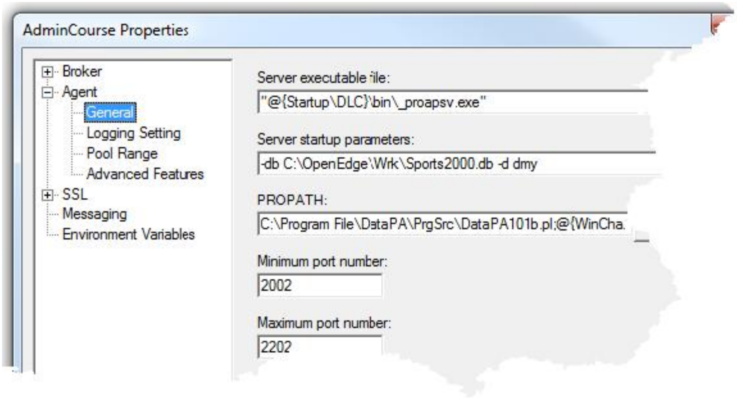
Maximum port number: 2202

© DataPA

Configuring the AppServer Agents

AppServer Agents, together with the AppServer Broker, make up the AppServer instance. They execute ABL procedures in response to DataPA client requests. To configure a basic AppServer Agent for DataPA, do the following:

1. Expand the Agent tree view and select General.
2. Enter any required start-up parameters in the server start-up parameters text box to connect databases and configure each AppServer agent.
3. Add the fully qualified path to the appropriate DataPA procedure library (see table below) to the PROPATH text box.



AdminCourse Properties

Server executable file: "@{Startup\DLC}\bin_proapsv.exe"

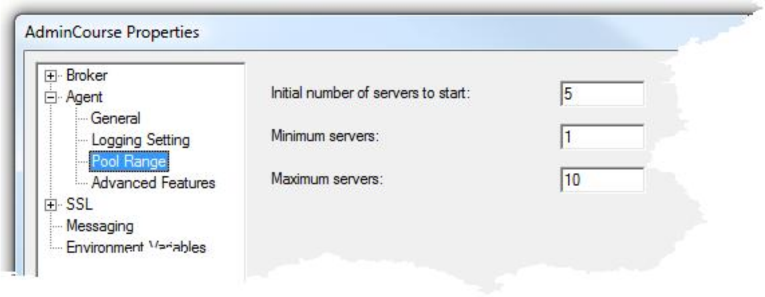
Server startup parameters: -db C:\OpenEdge\Wrk\Sports2000.db -d dmy

PROPATH: C:\Program File\DataPA\PrgSrc\DataPA101b.pl;@{WinCha

Minimum port number: 2002

Maximum port number: 2202

4. Select Logging Settings in the Agent treeview, and enter an appropriate name and location for the log file.
5. Choose Pool Range to define the range and quantity of Server Instances for this AppServer. For DataPA performance, you want to make sure that you set Maximum servers high enough to meet your reporting needs.

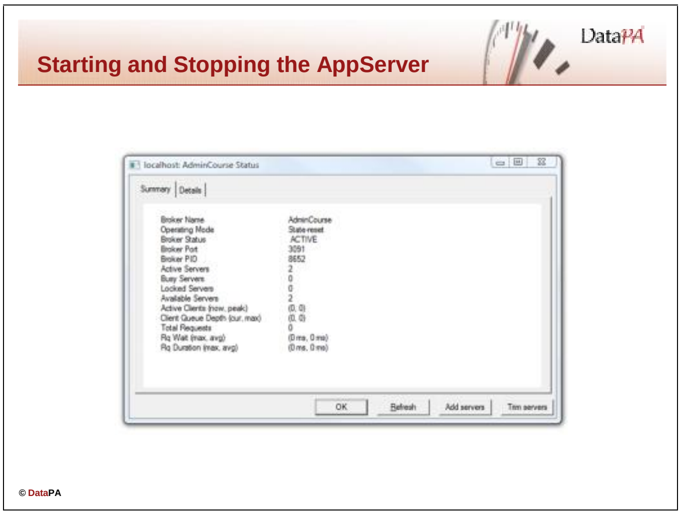


AdminCourse Properties

Initial number of servers to start: 5

Minimum servers: 1

Maximum servers: 10



Starting and Stopping the AppServer

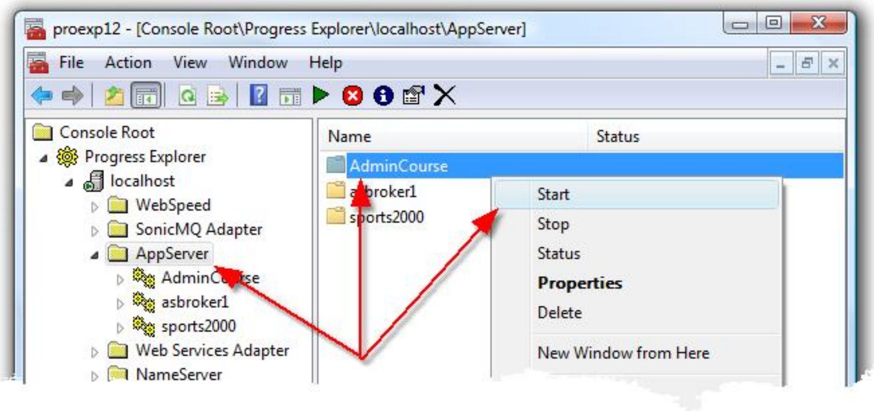
Broker Name	AdminCourse
Operating Mode	State-reset
Broker Status	ACTIVE
Broker Port	3091
Broker PID	8652
Active Servers	2
Busy Servers	0
Locked Servers	0
Available Servers	2
Active Clients (now, peak)	(0, 0)
Client Queue Depth (cur, max)	(0, 0)
Total Requests	0
Rq Wait (max, avg)	(0 ms, 0 ms)
Rq Duration (max, avg)	(0 ms, 0 ms)

© DataPA

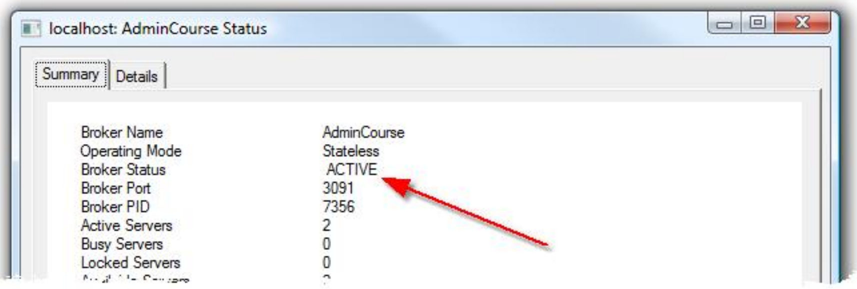
Manually starting the AppServer

Manual start is the default for the AppServer. Use the following steps to start an AppServer:

1. Choose the AppServer folder in the treeview. The list of available AppServer instances appears in the right pane.
2. Right-click the AppServer instance you want to start.
3. Choose Start.



4. Wait until the status message is "Started, please check status". Then wait a little more.
5. Choose the Status icon ℹ or Right-click the AppServer instance and choose Status.
   - This may take several attempts until the AppServer starts.
   - The status window displays with Broker Status ACTIVE when the AppServer has started successfully.

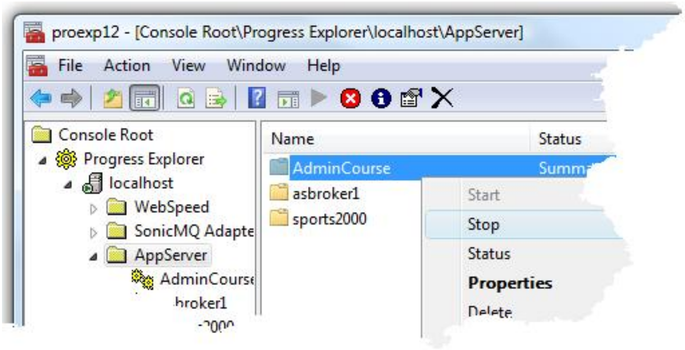


Broker Name	AdminCourse
Operating Mode	Stateless
Broker Status	ACTIVE
Broker Port	3091
Broker PID	7356
Active Servers	2
Busy Servers	0
Locked Servers	0

6. Choose OK in the Status window.
   The status display in the right pane reads “Summary/Status info fetched”.

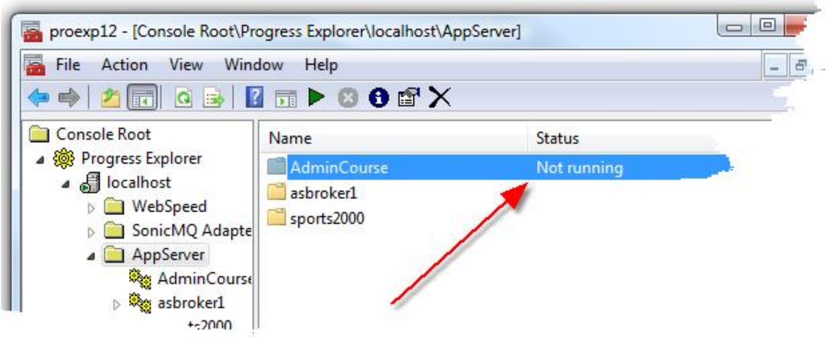
Stopping the AppServer

1. Use the following steps to stop an AppServer instance:
2. Choose AppServer folder in the treeview. The list of available AppServer instances displays in the right pane.
3. Choose the Stop icon ☒ or Right-click the AppServer instance and choose Stop.



Name	Status
AdminCourse	Summ
asbroker1	
sports2000	

4. The status display in the right pane reads “Stopped” and then “Not Running” when the AppServer stops.



Name	Status
AdminCourse	Not running
asbroker1	
sports2000	