

Reference Guide

D301490X412

April 2012

OpenEnterprise Fault Tolerance Network Reference Guide (V2.83)

Remote Automation Solutions

Website: www.EmersonProcess.com/Remote



Contents

| | | |
|----------|---|-----------|
| 1 | Fault Tolerance Configuration | 1 |
| 1.1 | Dedicated Fault Tolerance Network | 1 |
| 1.2 | How Fault Tolerance Works | 1 |
| 1.2.1.1 | The Journal Service | 1 |
| 1.2.1.2 | The Arbitrator Database | 1 |
| 2 | Fault Tolerance Task List | 2 |
| 2.1 | The Physical Network | 2 |
| 2.2 | Hosts File | 3 |
| 2.3 | Modify the Database Configuration File | 3 |
| 2.3.1 | Master Server Setup | 4 |
| 2.3.2 | Standby Server Setup | 4 |
| 2.4 | Build the Arbitrator | 5 |
| 2.4.1 | Monitoring a redundant session | 6 |
| 2.4.2 | Arbitrator Heartbeat Interval | 6 |
| 2.4.3 | Arbitrator Heartbeat Interval | 6 |
| 2.4.4 | Arbitrator Startup Timeout | 6 |
| 2.5 | Check the Arbitrator Database | 6 |
| 2.6 | Create a Redundant Session | 7 |
| 2.6.1 | How to create a Redundant Session on the Master Server | 7 |
| 2.6.2 | Session Type Page | 7 |
| 2.6.3 | Server Identification Page | 7 |
| 2.6.4 | Arbitrator Page | 8 |
| 2.6.5 | Configuring the Database Task | 9 |
| 2.6.5.1 | Add a Task | 9 |
| 2.6.5.2 | Define the Database Task | 10 |
| 2.6.5.3 | Set the Shutdown type | 11 |
| 2.6.5.4 | Advanced Shutdown Settings | 12 |
| 2.6.6 | Export the Session Settings from the Master Server | 13 |
| 2.6.7 | Importing the Session Settings to the Standby Server | 14 |
| 2.6.8 | Modifying the Fault Tolerant Session on the Standby Server | 15 |
| 2.6.8.1 | Server Identification Page | 15 |
| 2.6.8.2 | Arbitrator Key | 16 |
| 2.7 | Starting and Stopping a Redundant Session | 17 |
| 2.7.1 | Starting a Redundant Session | 17 |
| 2.7.1.1 | Scenario 1 - Starting a redundant session for the first time | 17 |
| 2.7.1.2 | Scenario 2 - Starting a Redundant Session on Subsequent Occasions | 18 |
| 2.7.2 | Stopping a Redundant Session | 18 |
| 2.8 | Causing a Manual Failover | 18 |
| 2.9 | OEPing INI File Settings | 19 |
| 3 | Index | 20 |

1 Fault Tolerance Configuration

For applications that require a high level of availability the OpenEnterprise product supports a Fault Tolerant configuration. In a Fault Tolerant system, an Active (Master) Server provides database services for OpenEnterprise Workstation clients, and a Standby Server keeps in step with it. Should a problem occur with the Active Server, the Standby is able to switch into Active mode, and continue processing database transactions where the previous Active Server left off.

1.1 Dedicated Fault Tolerance Network

When configuring a Fault Tolerant system, it is highly recommended to set up a dedicated Fault Tolerance Network between the Master and Standby Servers. The reasons for this are: -

1. Performance - Fault Tolerance related traffic is removed from the Workstation and RTU LAN.
2. Reliability.

The Fault Tolerance Network is also known as the JCP Network. Note that in order to provide true network redundancy, the redundant OpenEnterprise Servers must be fitted with dual Ethernet cards.

A Fault Tolerant OpenEnterprise system requires additional configuration to that required when running a Standalone OpenEnterprise session.

1.2 How Fault Tolerance Works

When OpenEnterprise is run as a Standalone system, only one database is required. However, in order to support a Fault Tolerant database, an extra Journal Service connection is employed for communication between the database on each Server, and an extra database called the Arbitrator is required.

1.2.1.1 The Journal Service

In order to implement a Fault Tolerant OpenEnterprise system, two identical instances of the database, are configured, one on each Server. The Master database transfers data to the Standby database in realtime using a dedicated journal stream. The journal stream is a TCP/IP connection identified by the service name jcp1.

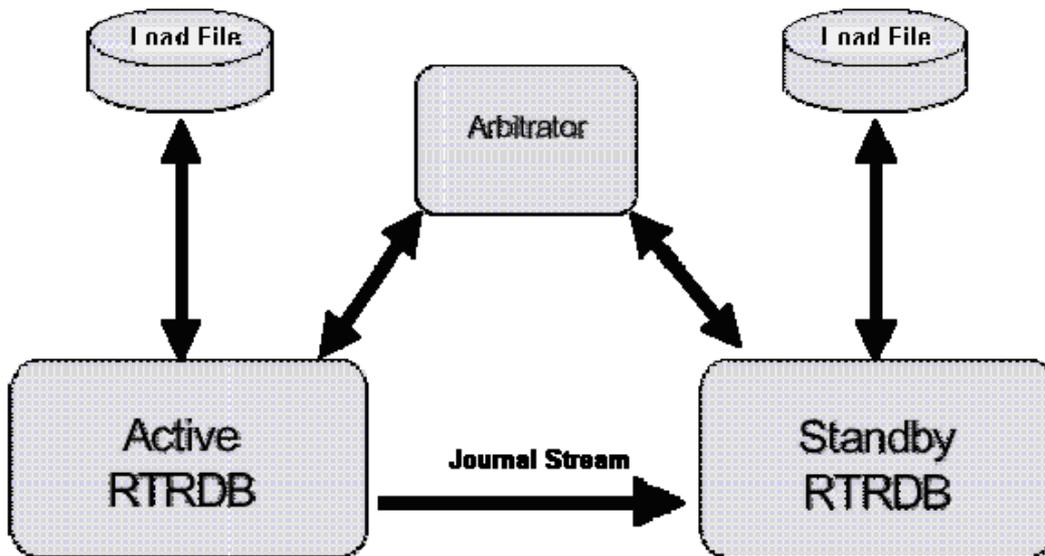
Note that in previous versions of OpenEnterprise this functionality was provided by the Journal Control Process (JCP) as a separate Windows task, but it has now become part of the main database process.

How does each database know whether it is to pass or receive database transactions to the other database? This is where the Arbitrator database comes in (see the next section).

1.2.1.2 The Arbitrator Database

The Arbitrator database uses server and system data to determine and control fault tolerance. It informs the database on each server which one is the currently Active database and which is the Standby. The main databases connect to the Arbitrator database via the *arbitrator1* TCP port.

The diagram below shows the relationship between the two main databases and the Arbitrator.



2 Fault Tolerance Task List

In order to configure a Fault Tolerant OpenEnterprise system with a dedicated Fault Tolerant Network, you will need to: -

1. Create the physical network
2. Modify each Server's Hosts file
3. Modify the Database Configuration file.
4. Build the Arbitrator Database
5. Check the Arbitrator database
6. Create a Redundant Session
7. Starting and Stopping a Redundant Session
8. Causing a Manual Failover
9. Set up OEPing if required

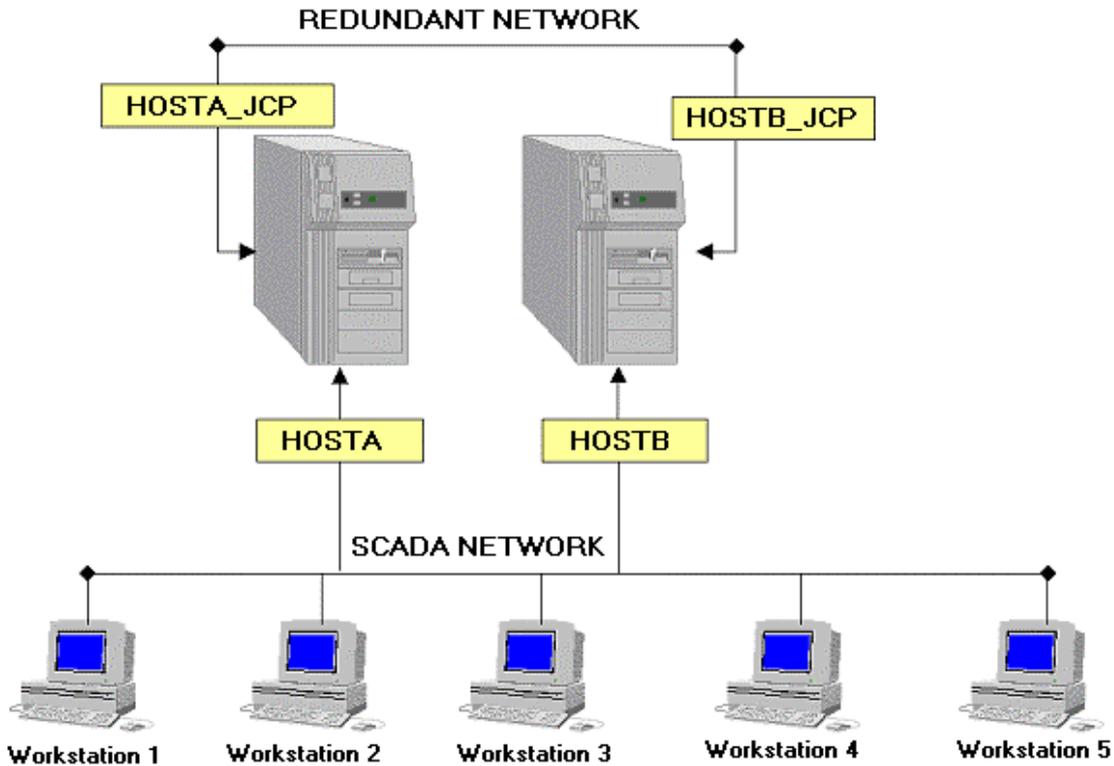
2.1 The Physical Network

The first step in configuring a Fault Tolerant Network for an OpenEnterprise system is to put one extra network card in each server to be used exclusively by the Fault Tolerant processes (JCP and Arbitrator). The Fault Tolerant network cards should have IP addresses that cannot be 'seen' by the SCADA network cards.

Then the Hosts file on each server should be edited to associate the local and remote IP address for each card with its respective Host name. The convention is to first decide on the Host names for the SCADA network (e.g. HOSTA and HOSTB), then to append an underscore followed by JCP to these Host names for the Redundant network Host names (e.g. HOSTA_JCP, HOSTB_JCP).

In the diagram below, the Active (Master) server has two network connections with Host names HOSTA and HOSTA_JCP respectively. The Standby server is configured identically to the Master with the exception that the two Host names are HOSTB and HOSTB_JCP.

The Redundant network is isolated from the SCADA network, and so provides network redundancy for the servers to exchange JCP (Journal Control Process) data. This ensures that if the SCADA network breaks, the two servers can still communicate with each other.



2.2 Hosts File

The Hosts file is found in the '%Windows%\System32\drivers\etc' directory. It matches IP addresses with Host names. Some of the server components are case sensitive so it is strongly recommended that only lower or uppercase characters be used in configuration text.

The following hosts file extract shows the required entries for both Master and Standby server.

```

11.100.0.100  HOSTA_JCP
11.100.0.101  HOSTB_JCP
101.100.0.100 HOSTA
101.100.0.101 HOSTB

```

} The JCP Network

} The Local Area Network

2.3 Modify the Database Configuration File

Fault Tolerance should be configured so that the RTRDB, when starting up, searches for the Arbitrator in the following order:

1. On the local host;

2. On the Fault Tolerance network;

This is done by ensuring that the 'ft' module in the database resource file (Poly.cfg) is set up correctly. The following are example Poly.cfg file entries for both Master and Standby.

2.3.1 Master Server Setup

The following is an extract from the Poly.cfg file on the Master Server. Note from the **arbitrator_service** resource the order in which the OpenEnterprise Database searches for an arbitrator as it starts up on the Main Server.

```
-----  
-- Fault tolerant settings...  
-----  
  
ft:db  
  
arbitrator_protocol = poly  
arbitrator_service = HOSTA_JCP:arbitrator1,HOSTB_JCP:arbitrator1  
journal_service = HOSTA_JCP:jcp1  
other_journal_service = HOSTB_JCP:jcp1  
enable_journalling = false  
arbitrator_run_isolated = true
```

2.3.2 Standby Server Setup

The following is an extract from the Poly.cfg file on the Standby Server. It mirrors the setup on the Master server, but the Journal Service and Other Journal Service names are reversed.

```
-----  
-- Fault tolerant settings...  
-----  
  
ft:db  
  
arbitrator_protocol = poly  
arbitrator_service = HOSTB_JCP:arbitrator1,HOSTA_JCP:arbitrator1  
journal_service = HOSTB_JCP:jcp1  
other_journal_service = HOSTA_JCP:jcp1  
enable_journalling = false  
arbitrator_run_isolated = true
```

2.4 Build the Arbitrator

The Arbitrator is a mandatory component within a Redundant OpenEnterprise Session. It is not required for a Standalone session. The Arbitrator is a special database that is used to decide which server should be the Master and which should be the Standby. At any one time only one arbitrator will be running on the Master Standby pair. Typically it will be running on the Master.

In order to configure an Arbitrator first start an empty arbitrator database by clicking on the Windows Start button and selecting: -

```
Start>Programs>OpenEnterprise>Arbitrator>Empty Arbitrator
```

Then start an SQL session to the arbitrator using: -

```
Start>Programs>OpenEnterprise>Arbitrator>SQL
```

At the SQL prompt type: -

```
SQL> include 'alltables';
```

```
SQL> shutdown;
```

The Arbitrator's schema is now initialised. Now start an Arbitrator to load the correct configuration using: -

```
Start>Programs>OpenEnterprise>Arbitrator>Arbitrator
```

The user should now edit the example database load file *Example.sql* found in the %INSTALLDRIVE%\Program Files\Bristol\OpenEnterprise\Database\ArbitratorSchema directory to add configuration specifics. (Change the name attribute to match the hostnames of the specific servers.)

Example Script for the Arbitrator:-

```
insert into oejcpcontrol ( name, heartbeat_interval, heartbeat_timeout,
startup_timeout, server_label) values ('ServerA:jcpl', SECONDS(30), 90,
120, 'A');
```

```
commit;
```

```
insert into oejcpcontrol ( name, heartbeat_interval, heartbeat_timeout,
startup_timeout, server_label) values ('ServerB:jcpl', SECONDS(30), 90,
120, 'B');
```

```
commit;
```

Note: The names used must match the journal_service name specified in poly.cfg and they are case sensitive. They should also match the host name values specified on the FaultTolerance key within the Settings Editor - (ThisHostName and OtherHostName).

Once satisfied with the edits the file must be saved to a different filename, keeping the SQL extension. The following assumes the file is called myarbitrator.sql.

Then start an SQL session to the arbitrator using the provided Arbitrator SQL shortcut.

At the SQL prompt type:-

```
SQL> include 'myarbitrator';
```

```
SQL> save into 'arbitrator.dat';
```

```
SQL> shutdown;
```

The user can optionally modify the:-

1. heartbeat_interval
2. heartbeat_timeout
3. startup_timeout values.

Adjust these values according to the application's requirements.

2.4.1 Monitoring a redundant session

The Session Manager will monitor and make available the status of both redundant servers. This state information is displayed in the Session Managers Status Bar and also written to the Redundancy table within the realtime database.

2.4.2 Arbitrator Heartbeat Interval

This determines how often the journal process will send a heartbeat to the arbitrator. It is recommended that this value should be at least a factor of three times less than the heartbeat_timeout value.e.g. SECONDS(30)

2.4.3 Arbitrator Heartbeat Interval

This determines how often the journal process will send a heartbeat to the arbitrator. It is recommended that this value should be at least a factor of three times less than the heartbeat_timeout value.e.g. SECONDS(30)

2.4.4 Arbitrator Startup Timeout

This value should be set to a count of seconds that corresponds to the amount of time it takes the system, from start-up to reach its idle state. Any errors heartbeat time-outs occurring within this period will be ignored.e.g. 12

2.5 Check the Arbitrator Database

The oejcpcontrol table in the Arbitrator database must show the Host names for each server in the correct case, as shown in the example below.

```
SQL> select name,server_label from oejcpcontrol;
```

```

+-----+-----+
| name           | server_label |
+-----+-----+
| 'HOSTA_JCP:jcp1' | 'A'         |
| 'HOSTB_JCP:jcp1' | 'B'         |
+-----+-----+

```

```
Query Done: 2 records selected
```

2.6 Create a Redundant Session

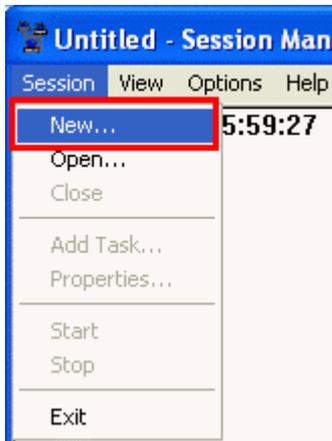
The settings for a redundant OpenEnterprise session are different on the Master and Standby Server. For this reason, we recommend the following procedure:-

1. Create the redundant session first on what will be the initial Master Server
2. Import that configuration to the other (Standby) Server using the Settings Editor
3. Make the necessary changes to the session for the other Server

Note that although server A is described as the initial Master and server B is described as the initial Standby, either can become the Master or Standby.

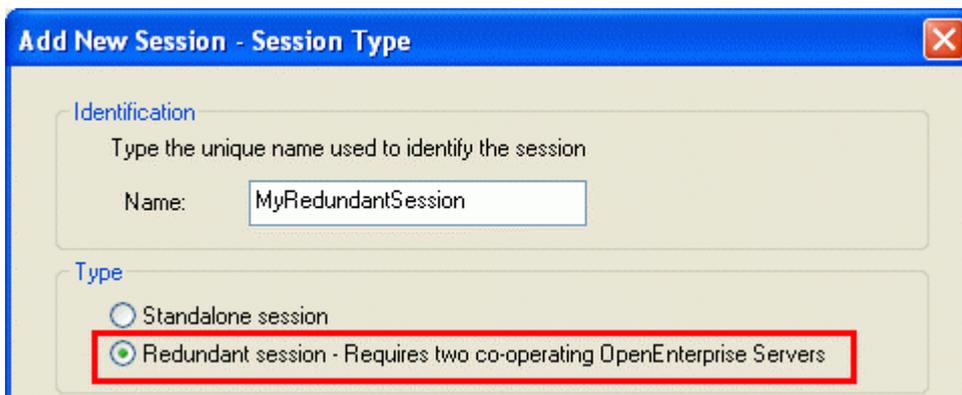
2.6.1 How to create a Redundant Session on the Master Server

Open the Session Manager and select the Session>New menu option.



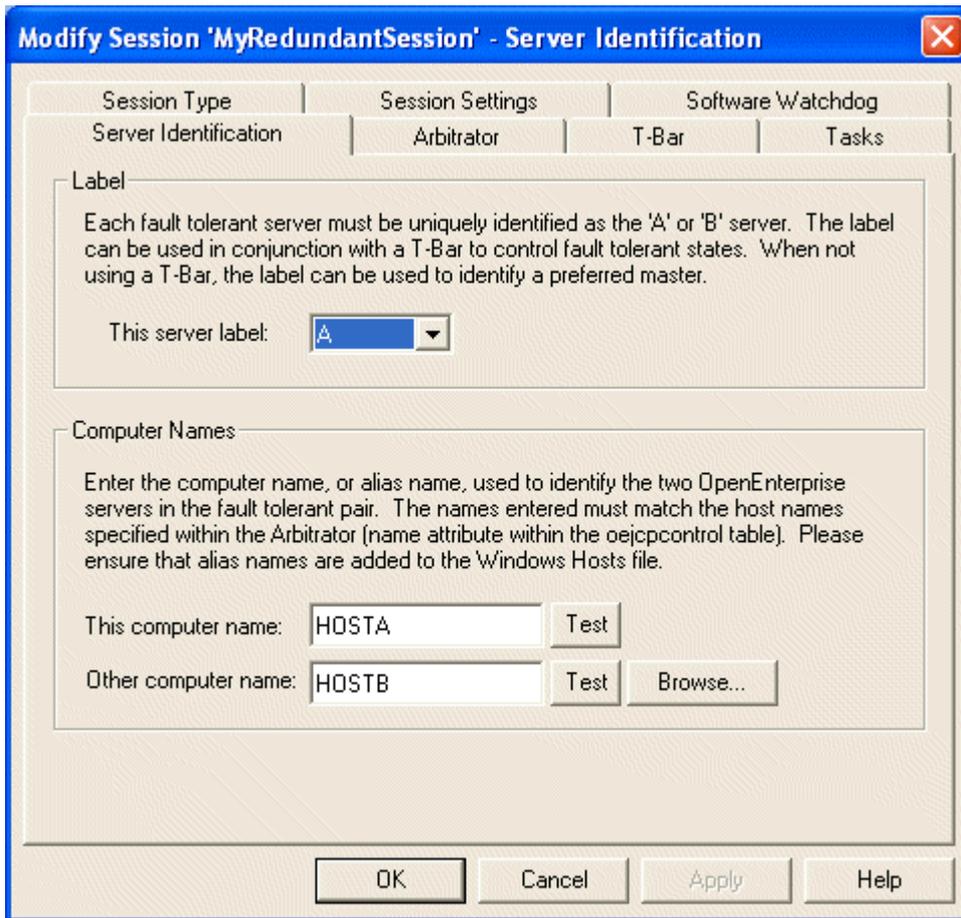
2.6.2 Session Type Page

This will open the Add New Session wizard. On the first page of the wizard you must select the 'Redundant session' option.



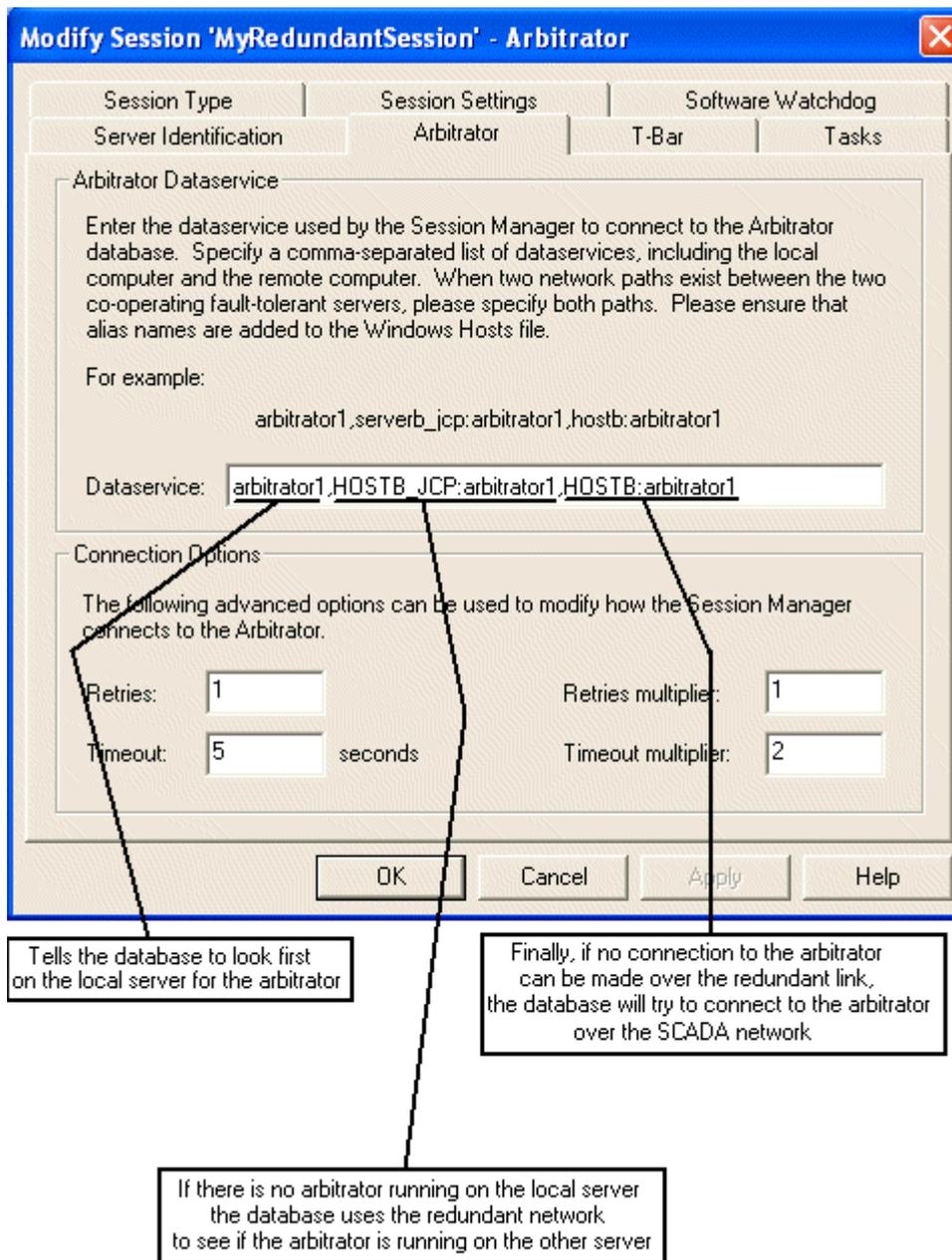
2.6.3 Server Identification Page

The Master Server should be set up with a machine label of 'A'. This computer name should be set to the name or alias of the Master Server (here 'HOSTA'). The other computer name should be set to the name or alias of the Standby Server (here 'HOSTB').



2.6.4 Arbitrator Page

The recommended data service string for connecting to the Arbitrator is shown below. The string defines the local arbitrator data service first, followed by two data services where the Session Manager may find the Arbitrator if it is running on the other Server. The first of these remote Arbitrator data services directs the Session Manager over the Server dedicated redundant network, the second directs the Session Manager over the normal SCADA network.

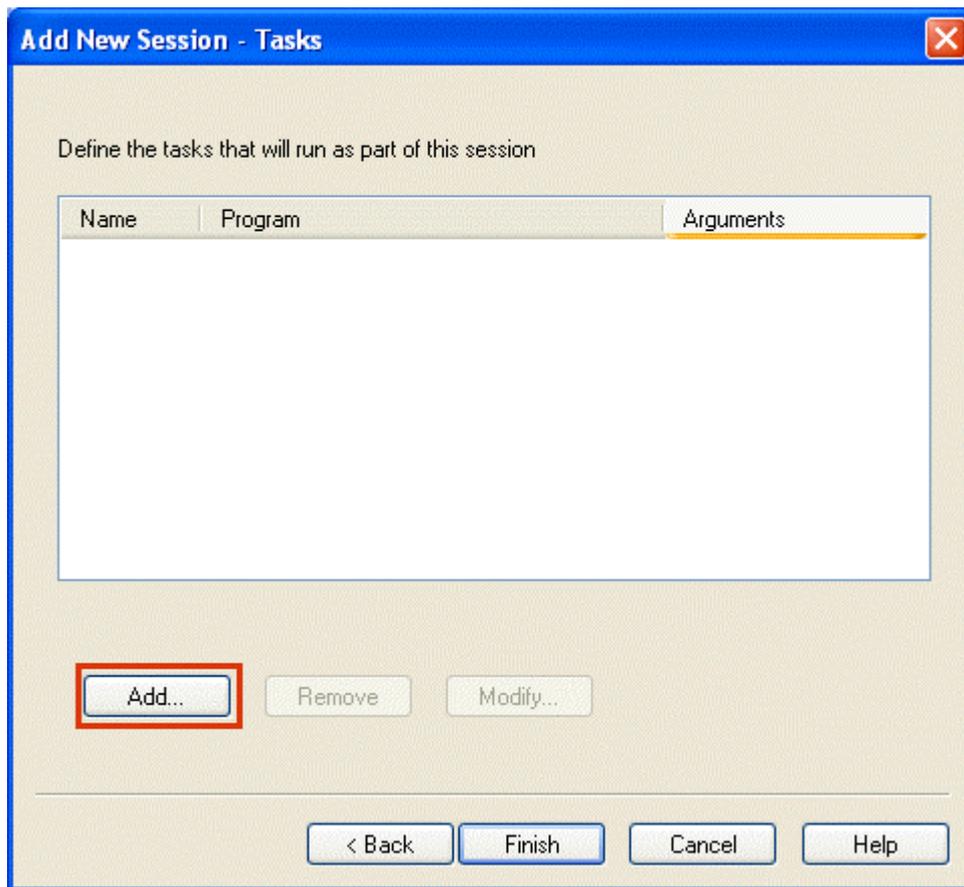


2.6.5 Configuring the Database Task

The Tasks page of the New Session Wizard enables us to set up the tasks that will run as part of the session. The main task of any session is the Database. Here is an example setup for the Database task.

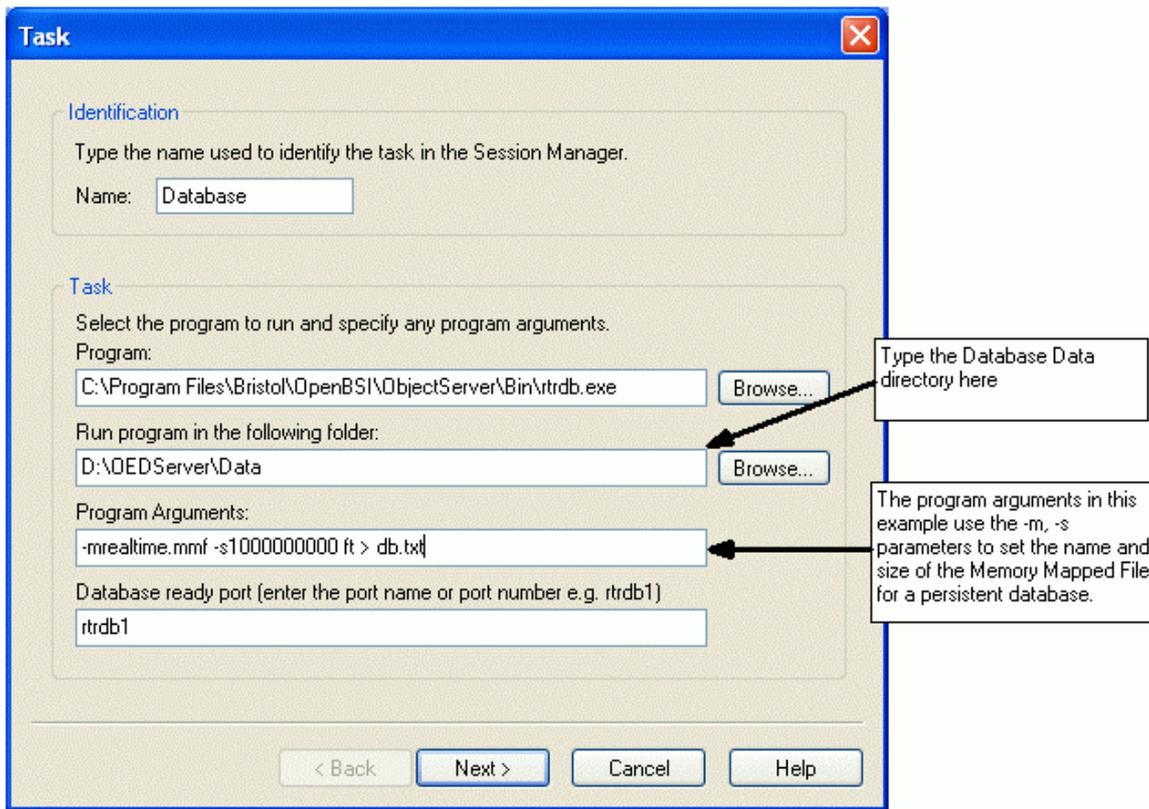
2.6.5.1 Add a Task

Firstly, click the **[Add]** button on the **Add New Session - Tasks** page.



2.6.5.2 Define the Database Task

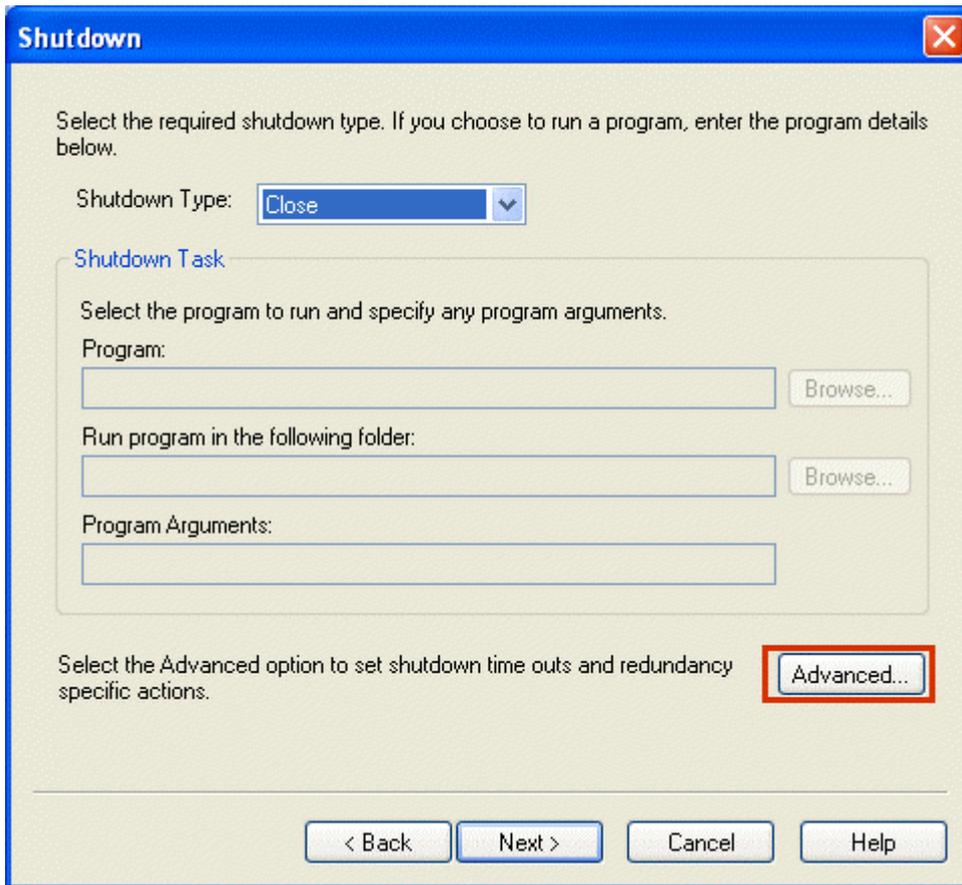
On the **Tasks** page type the task name and use the **[Browse...]** button to select the **rtrdb.exe** file. Then set the Database Data directory and the Program Arguments that are required. If you are running a persistent database you will need to use the **-m** and **-s** parameters to define a memory mapped file, as shown in the example below. For more information see the Configuring Persistence page in the Session Manager documentation.



2.6.5.3 Set the Shutdown type

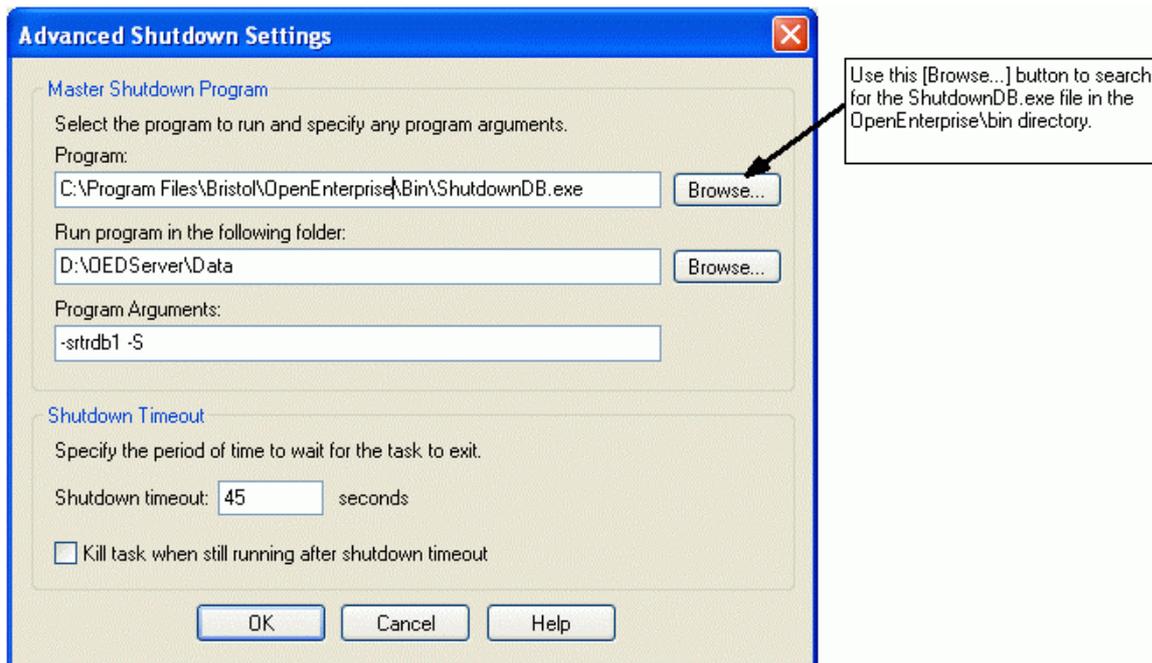
The Shutdown Type for the Database task should be set to **Close**. This means that if the Server is running the database in **Standby** mode it will close the database without saving it.

However, for a Redundant database, the Advanced Shutdown setting should always be used to specify the settings for a Database shutdown if the Server is running as the **Master**. So we must click the **[Advanced]** button when setting this up.



2.6.5.4 Advanced Shutdown Settings

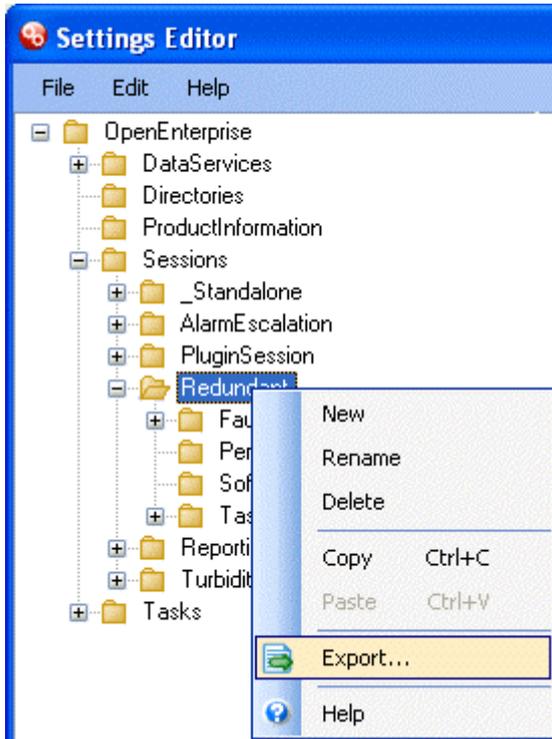
On the Advanced Shutdown settings page for a redundant database we need to point the Session Manager to the ShutdownDB executable. This will save the database before closing it when the database is running as Master on this Server.



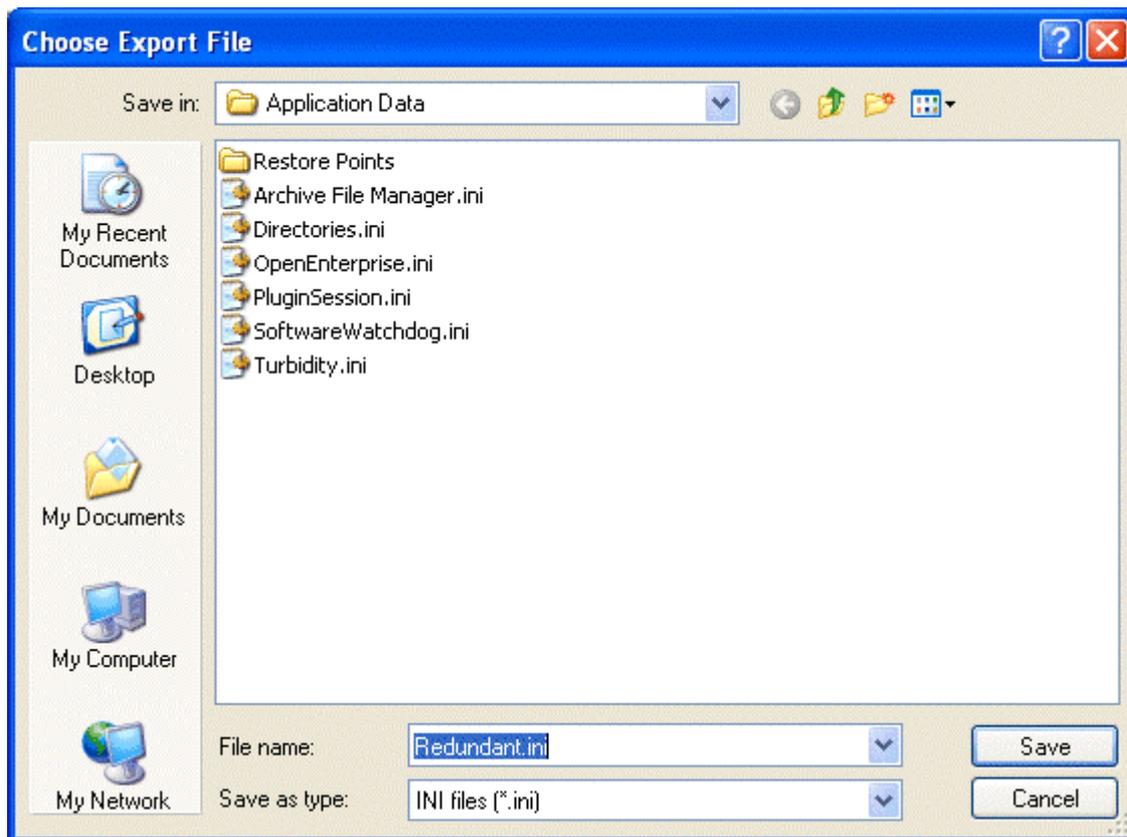
Once we have configured the Database Task we need to set up any other Sever tasks that we need for the Session. See the OpenEnterprise Help File for a list of possible Server tasks and a brief description of their purpose.

2.6.6 Export the Session Settings from the Master Server

Open the Settings Editor on the Master Server, select the key for the created session, then select the Export option from the context menu.

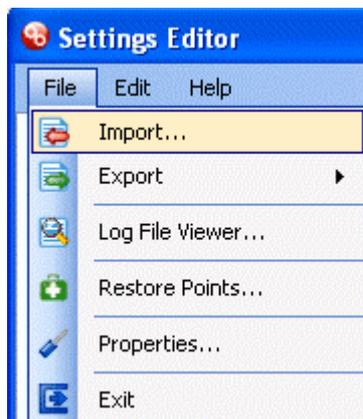


Save the file with the name provided, which will be the name of the session. Then copy it over to the Application directory on the Standby Server.

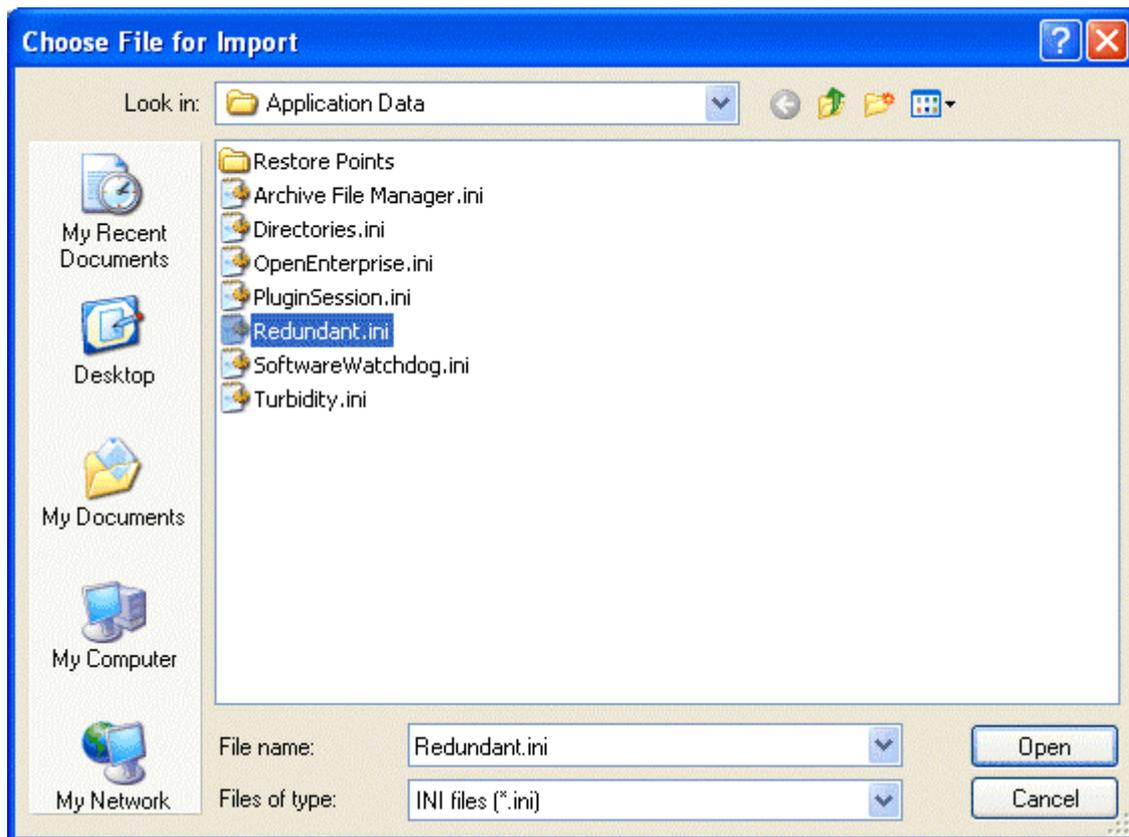


2.6.7 Importing the Session Settings to the Standby Server

Open the Settings Editor on the Standby Server, then select File>Import from the menu bar.



Locate the exported session that was copied over from the Master Server, and open it.

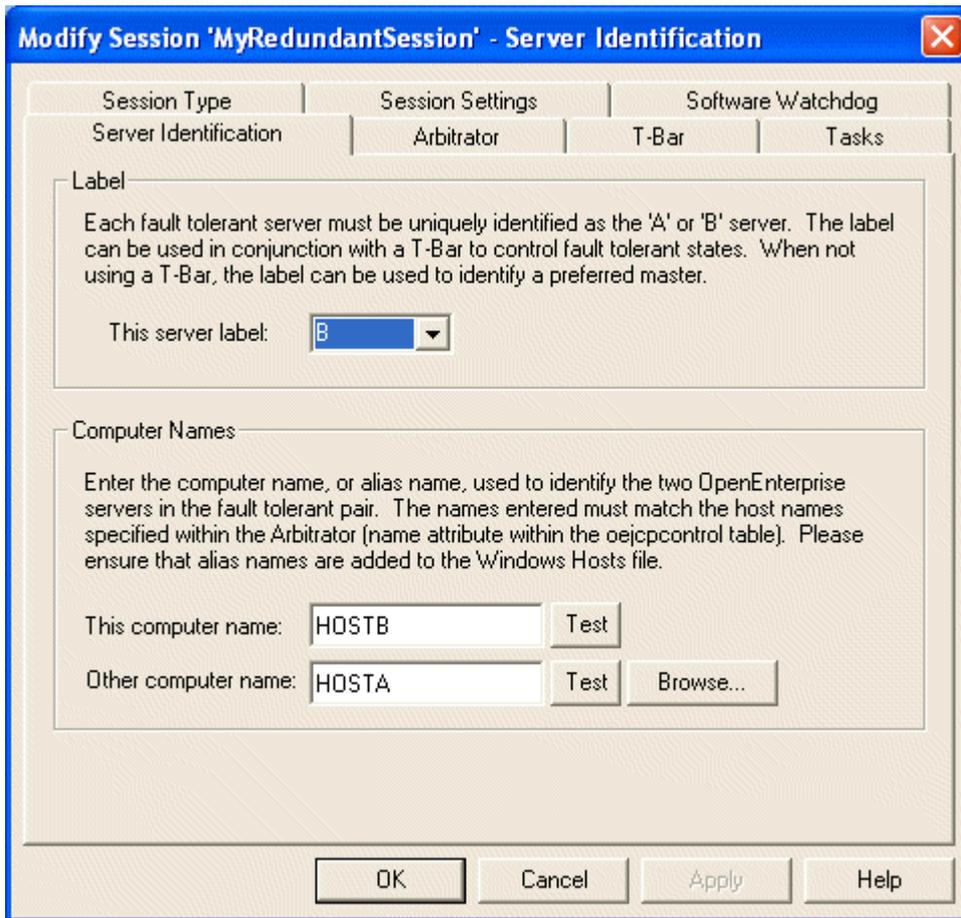


2.6.8 Modifying the Fault Tolerant Session on the Standby Server

After importing the session that was created on the Master Server onto the Standby Server, run the Session Manager on the Standby machine and open the imported session. The machine label and remote arbitrator data service settings now need to be modified for the Standby Server.

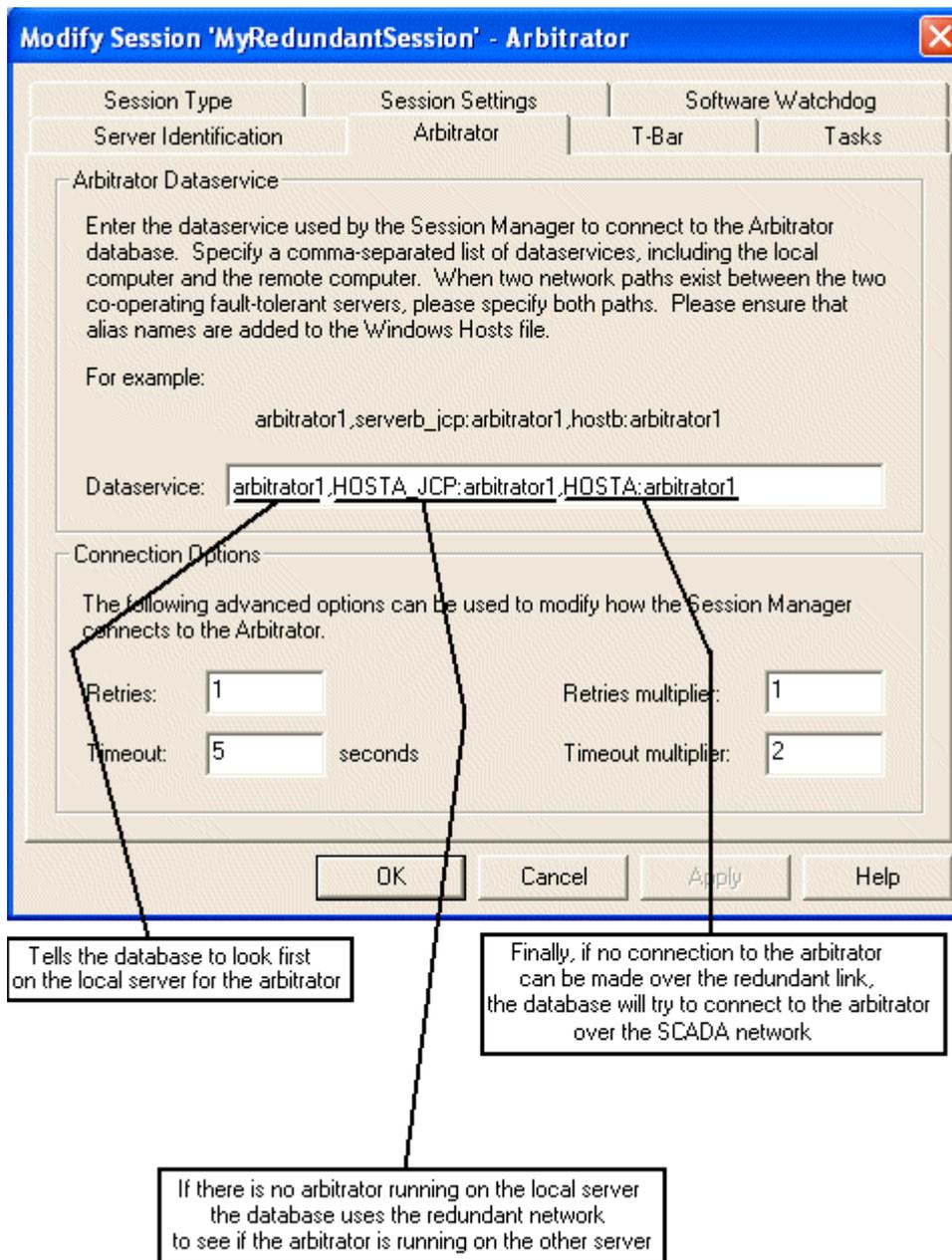
2.6.8.1 Server Identification Page

The imported session should now be modified so that the Standby has a machine label of 'B'. This computer name should be set to the name or alias of the Standby Server (here 'HOSTB'). The other computer name should be set to the name or alias of the initial Master Server (here 'HOSTA').



2.6.8.2 Arbitrator Key

The Dataservice value is now modified so that if the arbitrator is not running on the local machine, the database can connect to the arbitrator service on the other server (the original Master), first going through the redundant network, and then, if necessary, via the SCADA network.



2.7 Starting and Stopping a Redundant Session

In order to safely start and stop a redundant OpenEnterprise session we need to observe these simple rules.

2.7.1 Starting a Redundant Session

There are two possible scenarios when starting a redundant session.

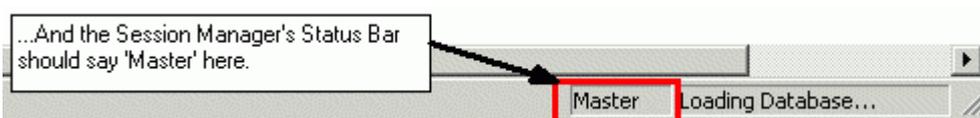
2.7.1.1 Scenario 1 - Starting a redundant session for the first time

- In this scenario, it is probably best to first start the Session on the Server that we have designated as Server A. This will initially be our **Master Server**.
- To do this, open the Session Manager on that Server and select the Session that you configured using Session>Open.

- Then select Session>Start.
- When the Master Server is properly up and running you will be informed in the Session Manager's Status Pane:-



And also in the Status Bar:-



- We can then start the session on the other Server that we designated as Server B. The Status Pane and Status Bar of the Session Manager will indicate that the database is loading. Then after a time it will inform us that the database is now in a **Standby State**. When the Servers are running as Master and Standby we have a redundant database.

2.7.1.2 Scenario 2 - Starting a Redundant Session on Subsequent Occasions

- We should first make doubly sure that neither of the redundant Servers is running an OpenEnterprise session.
- In this scenario we must always begin with the Server that was most recently the **Master Server**. We should open the Session Manager and start the session on this Server.
- Once the Master session reports that it is in **Master State**, we can then start the session on the Standby.

2.7.2 Stopping a Redundant Session

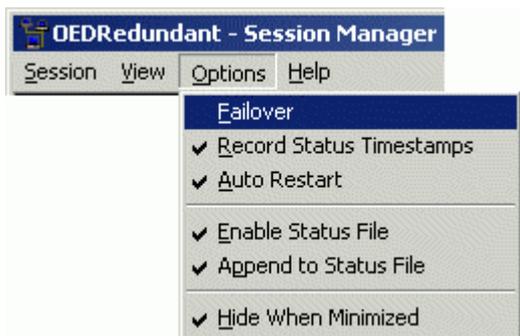
If we want to stop a redundant session, we have to first stop the **Standby session** from the Session Manager with the **Session>Stop** menu item in the Session Manager. The Standby Session will take a while to close down.

When the Standby session has stopped, we can then close the **Master** session that is running on the other Server by selecting the **Session>Stop** menu from the Session Manager on that machine.

2.8 Causing a Manual Failover

From time to time you may wish to cause a manual failover between the Servers. This means that the Master Server will stop and the current Standby Server will become the Master Server.

This can be achieved from either Server by selecting the Options>Failover item from the Session Manager's menu bar.



We strongly advise against using the Session>Stop menu item on the Master Server for this operation.

2.9 OEPing INI File Settings

The following shows the required OEPing.ini configuration for Master and Standby servers with host names of 'HOSTA' and 'HOSTB'. The OEPing.ini file should go into the Windows directory on each OpenEnterprise Server.

```
[HOSTA]

Hosts=HOSTA:0,Nexus:1,HOSTB:4

Arbitrator1=arbitrator1,HOSTB_jcp:arbitrator1,HOSTB:arbitrator1

ServerLetter=A

PingFrequency=10000

PingTimeout=50

PingRetries=1

RefreshInterval=10000

[HOSTB]

Hosts=HOSTB:0,Nexus:1,HOSTA:4

Arbitrator1=arbitrator1,HOSTA_jcp:arbitrator1,HOSTA:arbitrator1

ServerLetter=B

PingFrequency=10000

PingTimeout=50

PingRetries=1

RefreshInterval=10000
```

Finally, the **OEPing** component should be configured to run as a task in the Redundant Session. See the OEPing help file for more information.

3 Index

A

Arbitrator Registry Settings 8

D

Database Poly.cfg File 5

H

Hosts File 5

J

JCP Network Diagram 4

O

OEJCPCControl Table Settings 8

OEPIing INI File Settings 21

Overview 3

Reference Guide

D301490X412

April 2012

DISCLAIMER

Bristol, Inc., Bristol Babcock Ltd, Bristol Canada, BBI SA de CV and the Flow Computer Division, are wholly owned subsidiaries of Emerson Electric Co. doing business as Remote Automation Solutions ("RAS"), a division of Emerson Process Management. ROC, FloBoss, ROCLINK, Bristol, Bristol Babcock, ControlWave, TeleFlow and Helicoid are trademarks of RAS. AMS, PlantWeb and the PlantWeb logo are marks of Emerson Electric Co. The Emerson logo is a trademark and service mark of the Emerson Electric Co. All other marks are property of their respective owners.

The contents of this publication are presented for informational purposes only. While every effort has been made to ensure informational accuracy, they are not to be construed as warranties or guarantees, express or implied, regarding the products or services described herein or their use or applicability. RAS reserves the right to modify or improve the designs or specifications of such products at any time without notice. All sales are governed by RAS' terms and conditions which are available upon request. RAS does not assume responsibility for the selection, use or maintenance of any product. Responsibility for proper selection, use and maintenance of any RAS product remains solely with the purchaser and end-user.

Engineered and supported by:

Remote Automation Solutions,

Blackpole Road, Worcester, WR3 8YB, UK

Registered office: Meridian East, Leicester, LE19 1UX

Registered in England and Wales, Registration No. 00671801

VAT Reg No. GB 705 353 652

Emerson Process Management
Remote Automation Solutions
1100 Buckingham St
Watertown, CT 06795
T 1 (860) 945 2200
F 1 (860) 945 2278
www.EmersonProcess.com/Remote
binfo@EmersonProcess.com

Emerson Process Management
Remote Automation Solutions
Blackpole Road
Worcester, WR3 8YB
T 44 (0) 1905 856848
F 44 (0) 1905 856930
www.EmersonProcess.com/Remote
oedsupport@EmersonProcess.com

