

Adabas System Coordinator

Configuration

Version 8.2.7

December 2016

This document applies to Adabas System Coordinator Version 8.2.7.

Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.

Copyright © 2016 Software AG, Darmstadt, Germany and/or Software AG USA, Inc., Reston, VA, USA, and/or its subsidiaries and/or its affiliates and/or their licensors.

The name Software AG and all Software AG product names are either trademarks or registered trademarks of Software AG and/or Software AG USA, Inc. and/or its subsidiaries and/or its affiliates and/or their licensors. Other company and product names mentioned herein may be trademarks of their respective owners.

Detailed information on trademarks and patents owned by Software AG and/or its subsidiaries is located at <http://softwareag.com/licenses>.

Use of this software is subject to adherence to Software AG's licensing conditions and terms. These terms are part of the product documentation, located at <http://softwareag.com/licenses/> and/or in the root installation directory of the licensed product(s).

This software may include portions of third-party products. For third-party copyright notices, license terms, additional rights or restrictions, please refer to "License Texts, Copyright Notices and Disclaimers of Third-Party Products". For certain specific third-party license restrictions, please refer to section E of the Legal Notices available under "License Terms and Conditions for Use of Software AG Products / Copyright and Trademark Notices of Software AG Products". These documents are part of the product documentation, located at <http://softwareag.com/licenses> and/or in the root installation directory of the licensed product(s).

Use, reproduction, transfer, publication or disclosure is prohibited except as specifically provided for in your License Agreement with Software AG.

Document ID: COR-OS-VERSIONING-827-20161215

Table of Contents

Preface	v
1 Configuring Multiple Computers Without a Shared Configuration File	1
Firewall	2
Manual Group Configuration Example	3
Final steps for Manual Group Configuration	9
Troubleshooting	13

Preface



Note: This section uses a scenario of Data Archiving for Adabas as an example but the principles hold true in all situations.

Configuring Multiple Computers Without a Shared Configuration File

1 **Configuring Multiple Computers Without a Shared Configuration File**

■ Firewall	2
■ Manual Group Configuration Example	3
■ Final steps for Manual Group Configuration	9
■ Troubleshooting	13

By far the simplest way to configure across multiple computers is to allow automatic configuration and operation to take place during the runtime install on each computer. Automatic configuration and operation demands the (Adabas) configuration file is shared across all computers that are to participate together (in the same group). There are two ways the configuration file can be shared:

- One computer in the participating group “owns” the configuration file; all others reference it through the owning computer (using TCP/IP without Net-Work). This is the recommended mechanism.
- Net-Work is available so the file is accessible from all participating computers.

When sharing, the group configuration is constructed automatically in the shared file when the runtime is installed in each of the computers of the group. This is why sharing the file is so important. The automatically defined group is called SAGAUTO (SAG for Software AG and AUTO for automatic!).

Either of these two file-sharing mechanisms is the preferred way to install and run. As the runtime is installed on each computer a group is automatically built up in the shared configuration file. This group is SAGAUTO (SAG for Software AG and AUTO for automatic).

Firewall

Usually, the firewall in your organization understandably prevents processing across computers from functioning correctly by default. You must enable the appropriate ports in your firewall in order for this to work. The default ports are 53376-53399 in every computer – unless you change them yourself. If you change them you need to make an accurate note of the ports you define for every individual computer.

Manual Group Configuration Example

Some sites prefer not to share the configuration file even though multiple computers will need to collaborate as a group. Manual configuration is possible but it is far, far easier to allow the automation that comes with a shared configuration file to take place.



Note: It is the responsibility of the administrator to make sure the contents of the configuration file are correctly duplicated in every non-shared file. Even very small typing case differences will mean the duplicated manual configuration approach does not work. This is the nature of collaborative computing.

An example of manual duplicated configuration follows; we make an assumption that each computer uses its own copy of the configuration file. However, in a real situation it is feasible that some computers in the group are able to share the configuration file and some not. For the purposes of clarity within a complex topic it is simpler to explain assuming there is no sharing at all.

In this example the runtime is already installed on 2 computers (ukmcm001 and ukmcm002). This example starts from the point where:

- The runtime has been installed on each computer.
- Each has its own (local) configuration file.
- Normal automatic local configuration and operation took place in the install of the runtime in each computer (in isolation because the configuration file is not shared).
- The standard product demo runs successfully (locally) on each computer.
- The demo has been re-loaded in each computer ready to be reused.
- In summary, both computers are running successfully - but in isolation.

Manual duplicated configuration *must be repeated in every computer* where the configuration file is not shared. We start with the first computer (ukmcm001):

- 1. Define all other computers of the group (where the runtime is installed) in the ukmcm001 configuration file.
- 2. Define all these other computers to the group defined in ukmcm001.
- 3. Define the daemons running in all the other computers too.
- 4. Define the services running in all the daemons as well.

After doing this for the first computer the same thing has to be done in each other computer involved. This is the duplication of effort required for manual configuration. Every unshared configuration file must be fed with all the details of all the other computers in the same group. In this example there is only one other computer (ukmcm002)...

- 5. Define all other computers of the group (where the runtime is installed) in the ukmcm002 configuration file.
- 6. Define all the other computers to the group defined in ukmcm002.
- 7. Define the daemons running in all the other computers too.
- 8. Define the services running in all the daemons as well.

➤ **Here is the step by step walkthrough for doing this for ukmcm001 & ukmcm002 to operate as a group...**

- 1 Define all the other computers to the first of the group...

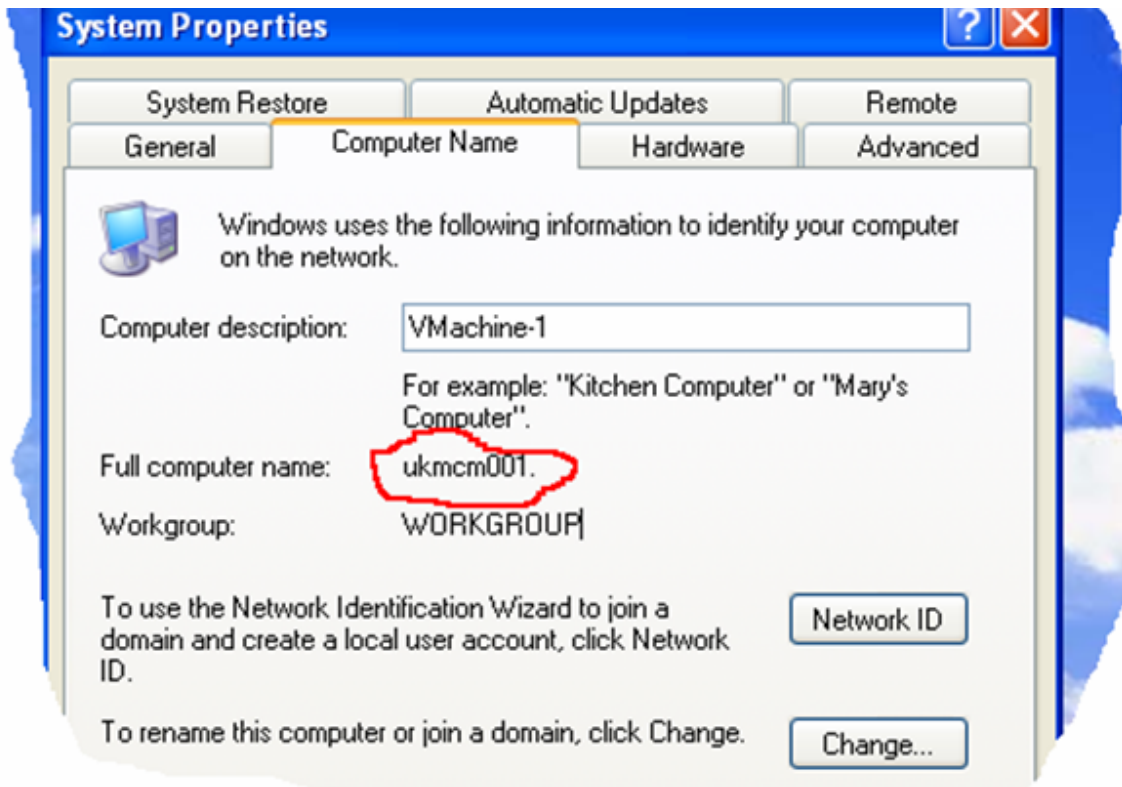
You will need to know the full computer name of all the computers to be involved in the group. As an example, in Windows you can find out the full computer name of any computer as follows:

Click **Start**.

Hover over **My Computer** then right-click.

Select **Properties**.

One of the tabs will be for **Computer Name**, select that tab and you will see...



In this example (above) the full computer name is ukmcm001 (ignore any ending period). The full computer name of the other computer is ukmcm002 for consistency (a picture is not shown). Now we have the computer names of all we can start the manual configuration for the first one (ukmcm001)...

Start an SMH browser session with ukmcm001.

Select **Adabas System Coordinator**.

Select **Computers** (Full list).

If the original install on ukmcm001 is correct then ukmcm001 will be visible here already.

Right-click **Computers** (Full list).

Select **Add computer**.

Enter a unique short name for this computer (this is the name that appears in the SMH tree).

Enter the full computer name in the **Fully qualified hostname** field.

Enter a text description if you wish to have one.

Select **Add computer**.

There are now two computers in the full computer list of ukmcm001. This means they are eligible to be selected in the group of collaborative computers.

2 Define all computers to the group

Select **Groups** within **Adabas System Coordinator**.

Select the **SAGAUTO** group.

You see the current local computer (ukmcm001) has already been automatically configured (you know this because the demo runs successfully).

Right-click the **SAGAUTO** group.

Select **Add computer to group**.

Select another computer from the drop-down list (this comes from the full list of computers, apart from the ones already defined to this group). In this case the computer to be selected is ukmcm002.

The standard ports will be filled in, we assume the standard ports have been used in each computer when you installed the runtime in each...if these were not used then you have to use the port numbers actually used for that computer when the runtime was installed.

3 Define the daemon running in the other computer

Right-click on the other computer just added to the group (ukmcm002).

Select **Add daemon to computer**.

Enter the short name **AUTO-1**.



Note: This name must match that which is defined in the other computer. AUTO-1 is created automatically, if you have modified the name in the other computer it must match exactly what is defined here.

Make sure start-up is selected as **Manual** (because this daemon does not run in ukmcm001, it is already running in ukmcm002 - we are just trying to tie up the duplicated definitions here, nothing more).

Set **Group services** as **Eligible**.

Enter a text description if you wish.

Select **Add daemon**.

4 Define the daemon running in the other computer

Right-click the **AUTO-1** daemon in ukmcm002.

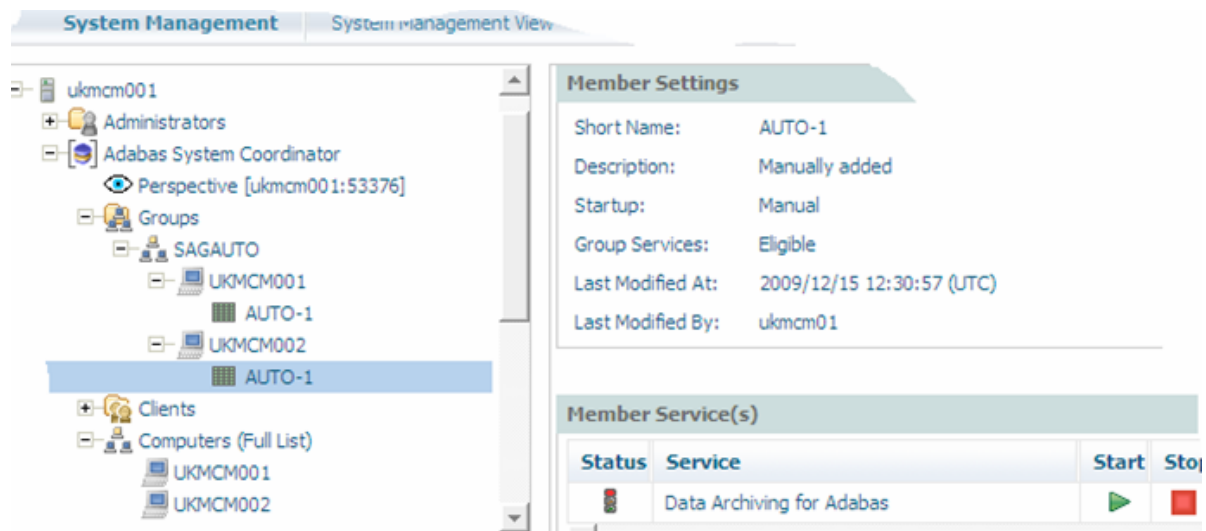
Select **Add Service to daemon**.

Select **Data Archiving for Adabas** (all the others are futures).

Click the **Add service** button.

Summary of ukmcm002 added to the configuration of ukmcm001

At this point we have only done part of the job. We have defined the details of the second example computer (ukmcm002) into the first (ukmcm001). Having done this we should see...



In the diagram above you can see...

- The Adabas System Coordinator configuration for ukmcm001 (only).
- The other computer (ukmcm002) has been added to the full list.
- This allows ukmcm002 to be added to the existing (automatic) SAGAUTO group (which now covers ukmcm001 and ukmcm002).
- The daemon of ukmcm002 has been defined too.
- and the archiving service within it.

Remember, all this work is *only* needed when shared configuration is not used. At this stage only a part of the duplication is achieved. Now we need to make reciprocal definitions in ukmcm002 (and all other computers after that if there are more)...

5 Define all the other computers to the next...

Start an SMH browser session with ukmcm002.

Select **Adabas System Coordinator**.

Select **Computers** (Full list).

If the original install on ukmcm002 is correct then ukmcm002 will be visible here already.

Right-click **Computers** (Full list).

Select **Add computer**.

Enter a unique short name for this other computer (ukmcm001: this is the name that appears in the SMH tree)..

Enter the full computer name in the **Fully qualified hostname** field.

Enter a text description if you wish to have one.

Select **Add computer**.

There are now two computers in the full computer list of ukmcm002. This means they are eligible to be selected in the group of collaborative computers.

- 6 Define all other computers to the collaborative group

Select **Groups** within **Adabas System Coordinator**.

Select the **SAGAUTO** group.

You see the current local computer (ukmcm002) has already been automatically configured (you know this because the demo runs successfully).

Right-click the **SAGAUTO** group.

Select **Add computer to group**.

Select another computer from the drop-down list (this comes from the full list of computers, apart from the ones already defined to this group). In this case the computer to be selected is ukmcm001.

The standard ports will be filled in, we assume the standard ports have been used in each computer when you installed the runtime in each...if these were not used then you have to use the port numbers actually used for that computer.

- 7 Define the daemon running in the other computer

Right-click on the other computer just added to the group (ukmcm001).

Select **Add daemon to computer**.

Enter the short name **AUTO-1**.



Note: This name must match that which is defined in the other computer. AUTO-1 is created automatically, if you have modified the name in the other computer it must match exactly what is defined here.

Make sure start-up is selected as **Manual** (because this daemon does not run in ukmcm002, it is already running in ukmcm001 - we are just trying to tie up the duplicated definitions here, nothing more).

Set **Group services** as **Eligible**.

Enter a text description if you wish.

Select **Add daemon**.

- 8 Define the archiving service running in the other computer

Right-click the **AUTO-1** daemon in ukmcm001.

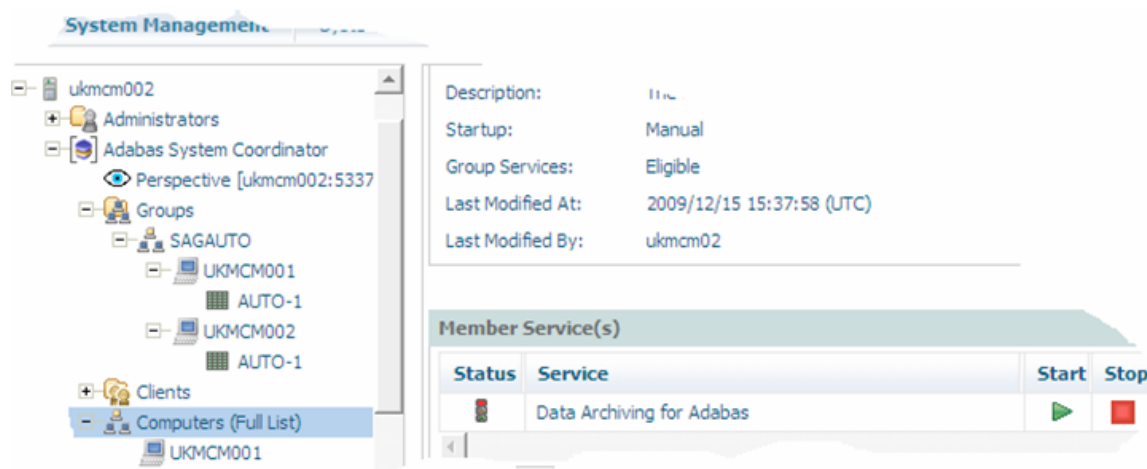
Select **Add Service to daemon**.

Select **Data Archiving for Adabas** (all the others are futures).

Click the **Add service** button.

Summary of ukmcm001 added to the configuration of ukmcm002

We have defined the details of the first example computer (ukmcm001) into the second (ukmcm002). Having done this we should see...



In the above you can see...

- The Adabas System Coordinator configuration for ukmcm002 (only).

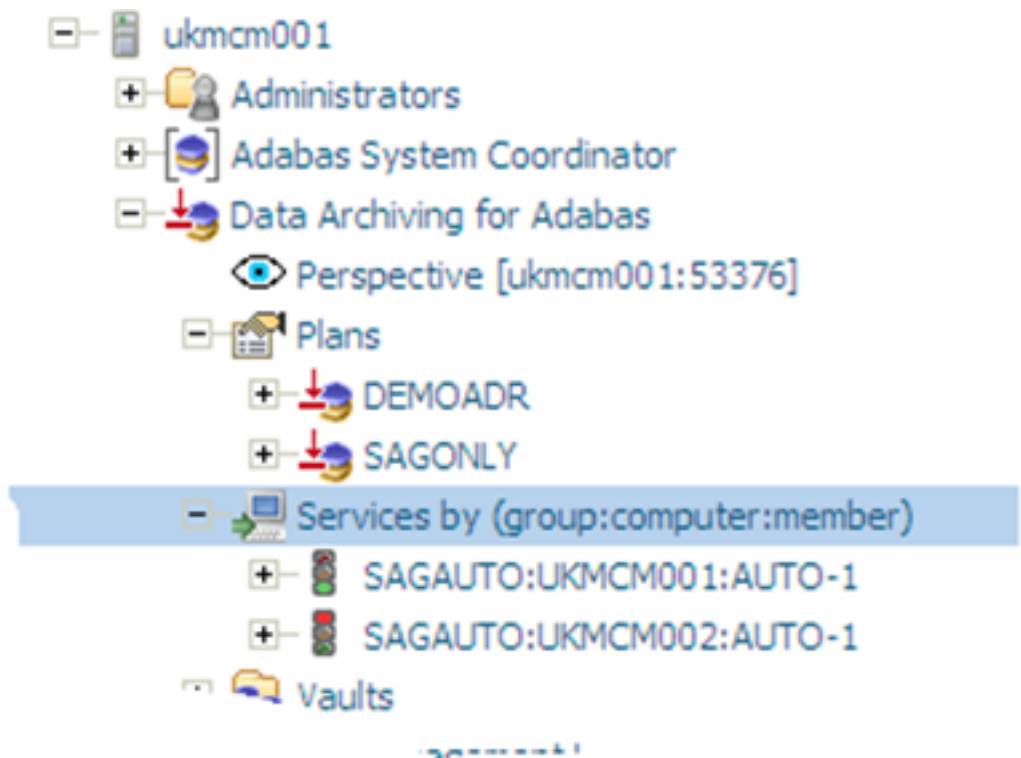
- The other computer (ukmcm001) has been added to the full list.
- This allows ukmcm001 to be added to the existing (automatic) SAGAUTO group (which now covers ukmcm001 and ukmcm002).
- The daemon of ukmcm001 has been defined too.
- and the archiving service within it.

Remember, all this work is only needed when shared configuration is not used.

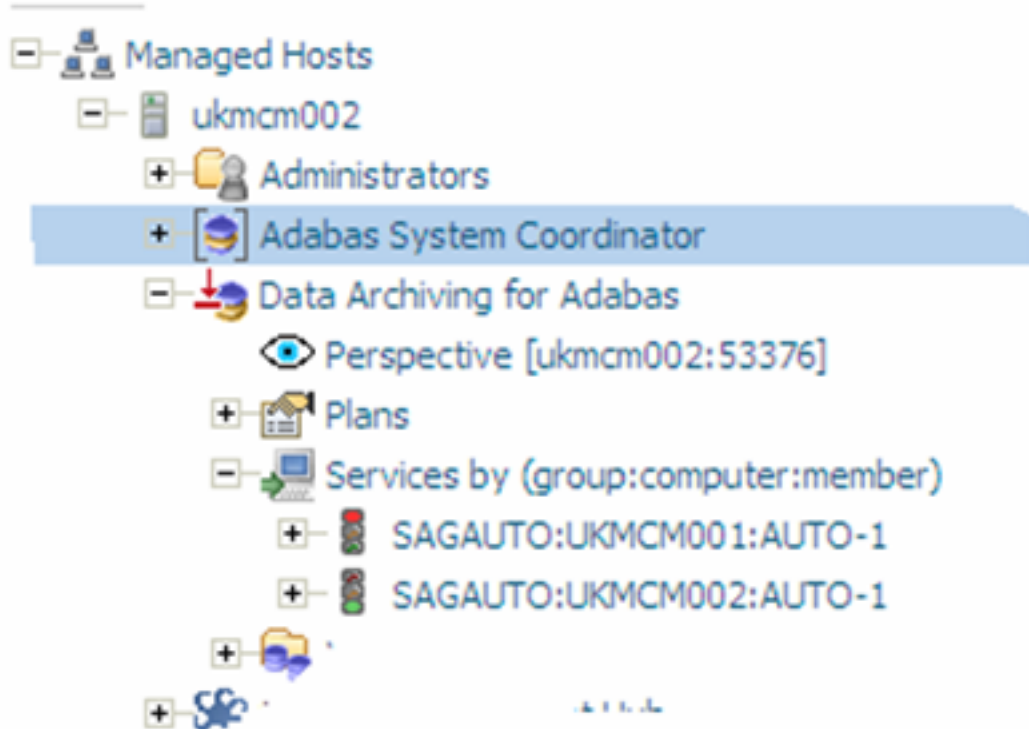
Final steps for Manual Group Configuration

In the example above we have successfully defined two computers (ukmcm001 and ukmcm002) in a reciprocal group called SAGAUTO across two configuration files. These definitions must be made twice because there isn't a single shared configuration file.

Now it should be possible to see Data Archiving for Adabas activity monitor displays for both computers in SMH. However, an activity monitor display (refreshed since the configuration changes) in ukmcm001 shows (below) that its own activity monitor is active (green light) but the activity monitor in the other (ukmcm002) computer is not active (red light)...



Similarly, the same situation (mirror image) happens in ukmcm002....



The problem appears to be a firewall issue (neither computer is able to ping the other). The firewall is temporarily disabled in both. But the problem persists. Something else is wrong too.

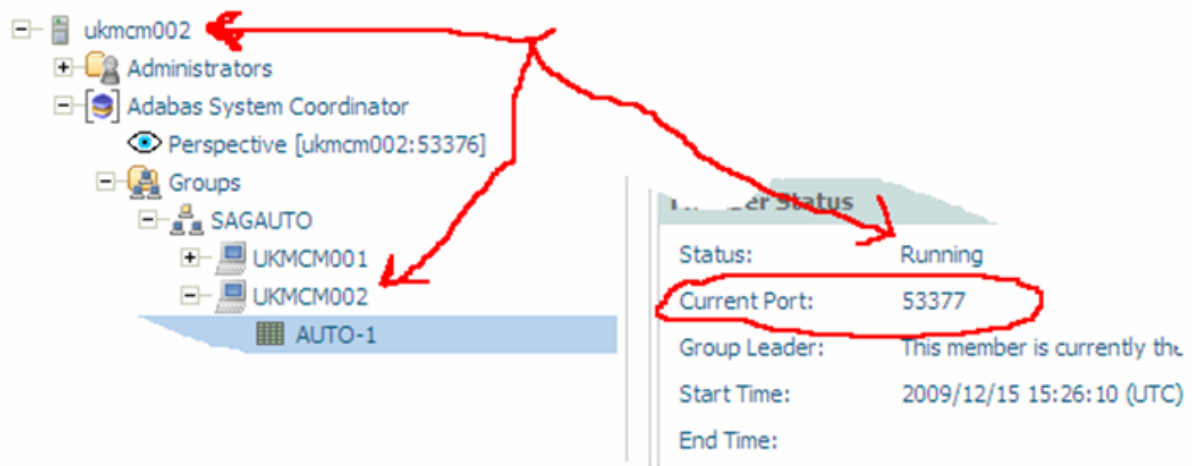
The second problem is a by-product of configuring manually. When each daemon starts up (automatically) in each computer it is assigned a port dynamically from the available range (by default this range is 53377 through 53399 inside each computer). Without automatic shared configuration it is not possible for each computer to find out which ports are being used in the other computers at any moment. So a further manual step is needed, as follows:

- a. Find out which port each running daemon (in each computer) is using.
- b. Define that specific port number in each manual configured daemon, in all computers.

By now it is becoming clear why auto-configuration with a shared file is by far the easiest option, to say the least! With a shared configuration file none of this manual effort is needed at all. To manually find out the port number allocated to each running daemon do the following:

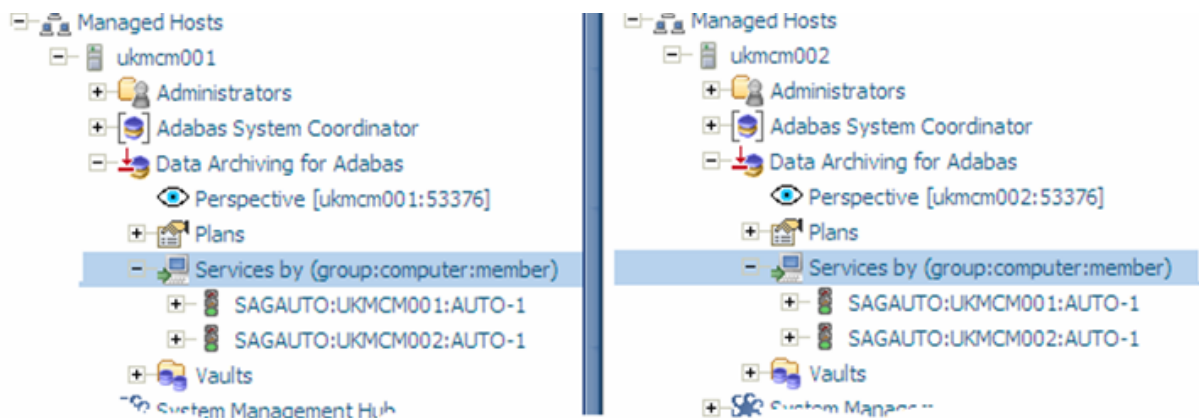
- a. For each computer where the runtime is installed (ukmcm001 & ukmcm002 in the example)...do all of the following...
- b. Start an SMH session.
- c. Go to the point in the SMH tree (shown in the diagram below) to note the port (for the computer being viewed); add port number into the reciprocal definition of this daemon in the configuration of all computers...

- d. In our example, default ports have been used so the same port number applies in all cases, but that may not be the case!



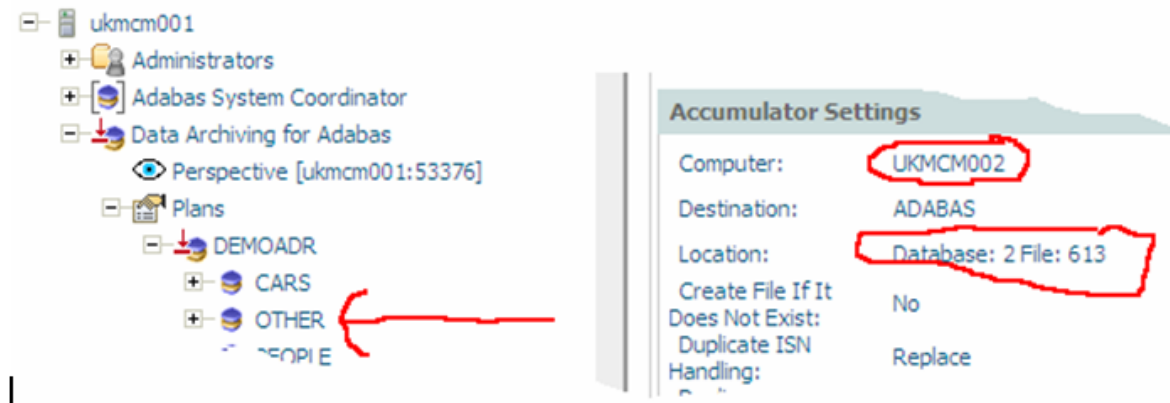
Once these reciprocal port definitions are made then the duplicated group definitions in each computers configuration file will be in a position to recognize each other. And, with the firewall issues out of the way they should be able to make contact.

You can test this by going into **Data Archiving for Adabas** in an SMH session in each computer. Previously the non-local archiving services were showing as inactive, even though we know they are actually running. Now (with refresh) we should see that they all show as active in each computer (SMH browser) session. This shows that the groups are successfully connected, as follows...

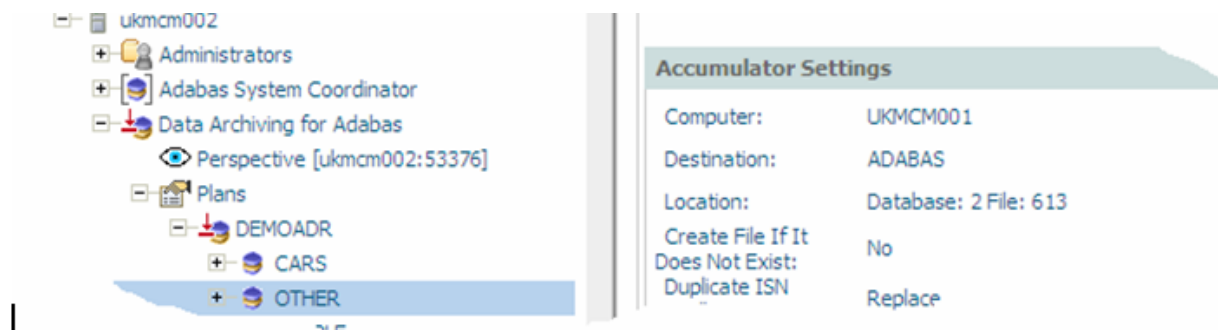


In the picture (above) we can see snippets of separate SMH sessions with ukmcm001 and ukmcm002 (our example computers). Now that the manual group definitions reciprocate, and the firewall obstacle is removed the daemons running in each computer are visible in every collaborating computer.

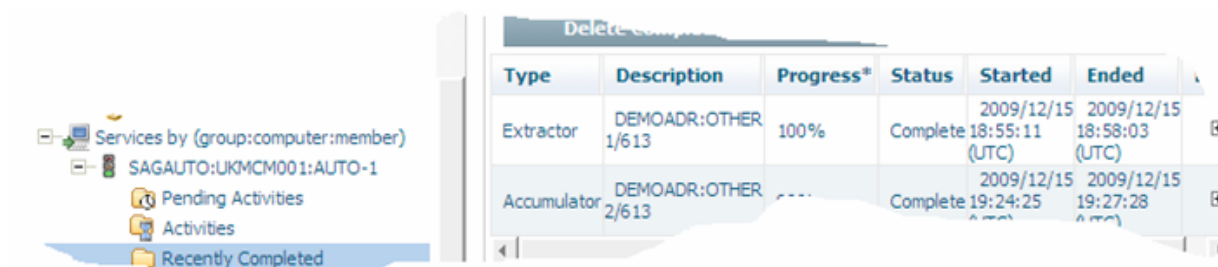
Now we can simply adjust the demo archiving configuration to do archiving across computers. One part of the demo is the OTHER action within the DEMOADR plan. Normally this is configured to run the extract and accumulate in the same (local) computer, but now we can configure it to run across computers – both ways!!!! Here is the modified configuration in ukmcm001 showing that the new destination for OTHER in the ukmcm001 configuration is Adabas database 2 file 613 in ukmcm002 (the extractor is unchanged and runs in ukmcm001)...



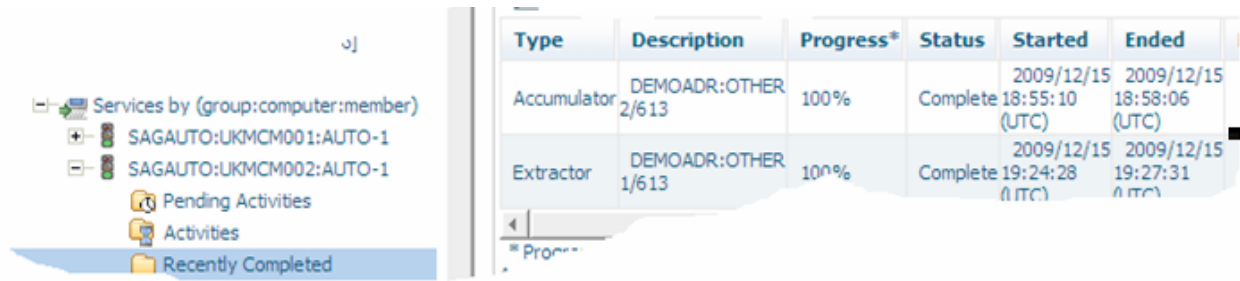
Similarly, the equivalent OTHER action in ukmcm002 will now accumulate into database 2 file 613 in ukmcm001 (so we have two-way archive actions):



Now we can run the action (the extractor end) in each computer and see that they run across the two computers, reciprocally as you can see in the activity monitor display of completed activities in ukmcm001...



...and ukmcm002...



Type	Description	Progress*	Status	Started	Ended
Accumulator	DEMOADR:OTHER 2/613	100%	Complete	2009/12/15 18:55:10 (UTC)	2009/12/15 18:58:06 (UTC)
Extractor	DEMOADR:OTHER 1/613	100%	Complete	2009/12/15 19:24:28 (UTC)	2009/12/15 19:27:31 (UTC)

Troubleshooting

When connecting processes across computers many confusing problems can occur. These problems are greatly minimized when sharing the configuration file because configuration and operation happens automatically; usually meaning the only obstacle is the firewall. Manual configuration is far more error prone. Here are some suggestions for problems that you may experience:

- Components cannot communicate (nothing seems to be happening). The firewall on some or all computers is obstructing communication.

Either temporarily disable the firewall on each computer or enable the ports that are used in each.

- Components outside of local computers seem to be inactive (red lights etc). The firewall on some or all computers is obstructing communication.

Either temporarily disable the firewall on each computer or enable the ports that are used in each.

- Firewall issues are taken care of but things still don't hook up.

Names of all reciprocal things must be exactly the same on all computers, case-sensitive – if they are not then they will not hook up.

- Confusing things are happening on unpredictable computers.

Computer full names are not unique within your network.

- A new action is added to the DEMOADR plan but it isn't working across computers.

The same definition needs to be made in the configuration file of all computers.

