iSeries

IBM

Subsystem configuration

Experience Report

iSeries

Subsystem configuration

Contents

Subsystem Configuration

The default subsystem configuration shipped with i5/OS(R) is a basic subsystem configuration that works well for small systems. However, as the number of users increases on the system, it is desirable to split the work into multiple subsystems to better manage the work on the system.

This experience report includes the following information:

Why consider multiple subsystems?

i5/OS(R) is shipped with a standard set of subsystems. The following subsystems are necessary for supporting user work:

- QBASE **or** QCTL, QINTER, and QCMN
- QSYSWRK
- QSERVER
- QUSRWRK

Interactive users typically run in QBASE or QINTER; which one you use is determined by the controlling subsystem system value (QCTLSBSD). This experience report assumes you are using QINTER as your interactive subsystem.

Server jobs run in QSYSWRK, QUSRWRK, and QSERVER. The daemon jobs (the server jobs that route work requests to waiting prestart jobs) typically run in QSYSWRK and QSERVER, while the prestart server jobs that perform work on behalf of users typically run in QUSRWRK, although some of the prestart jobs also run in QSERVER.

This experience report focuses on the users that run in QINTER and QUSRWRK. As the number of users on the system increases, a single subsystem for a set of work is often insufficient.

By dividing your users into multiple subsystems you gain several advantages, including the following:

- Improves manageability of the work on the system due to better intellectual control over what work is running in what subsystems. For example, for server jobs, you may want to isolate all of the database server jobs to one subsystem, the remote command server jobs to a different subsystem, the DDM server jobs to yet a different subsystem, etc. This allows you to understand of the type of work being done in the various subsystems.
- Provides the ability to disallow sets of users to access the system at certain periods of time. For example, if every Friday afternoon you must bring the system to the restricted state for backup purposes, you can gradually take users offline by ending one subsystem at a time.
- Improves scalability and availability. By having a single subsystem do work for fewer users, the subsystem is less busy and can be more responsive to the work requests it handles.
- Improves error tolerance by spreading the work across multiple subsystems; this is particularly important for interactive subsystems. For example, if a network failure occurs and many sessions enter into device recovery, the interactive subsystem can become very busy with the device recovery processing. This is because the subsystem job is the central job for managing device recovery. If hundreds of devices are affected by a network failure, they are recovered by the subsystem a few at a time; such processing can negatively impact the ability for users to sign on or sign off the system. By splitting the users into

multiple interactive subsystems, you can have more than one subsystem job manage the device recovery processing. Limiting the number of devices in one subsystem provides improved availability and scalability of the interactive subsystem.

- Improves interactive subsystem startup time. When an interactive subsystem is started, it attempts to allocate all the devices defined by the workstation entries added to the subsystem description. As the number of devices increases, subsystem startup time may increase. You can keep subsystem startup times shorter by subdividing the work across multiple subsystems.
- Provides additional options for performance tuning. Several performance attributes, such as run priority, time slice, default wait, maximum CPU and others are performance attributes specified in the class description. Storage pools and maximum jobs are specified on the subsystem description. You can tune performance by using multiple routing entries with appropriate classes within a single subsystem, or you can simply set up multiple subsystems with a small number of routing entries. Either way, you give more system resources to users doing business critical work while fewer system resources are available for users doing work that is less important. Which approach you use depends upon your requirements. The detailed information later in this experience report uses multiple subsystems with simpler configurations.

Server subsystems

The default subsystem configuration shipped with i5/OS(R) is a basic subsystem configuration that works well for small systems. However, as the number of users and amount of work increases on the system, you may want to split the work into multiple subsystems to better manage the work on the system.

This experience report includes the following information:

"Multiple subsystem considerations"
This section describes examples where using multiple subsystems is a good idea and describes some things to be aware of when creating and configuring subsystems.

"Server subsystem configuration" on page 3
This section describes how to perform subsystem configuration for server jobs.

"Server subsystem defaults" on page 10
This section describes how to modify server subsystem default values.

"User-defined server subsystems" on page 13
This section describes how to create a user-defined server subsystem.

Multiple subsystem considerations

Server jobs are jobs that run continuously in the background on the iSeries(TM) server waiting for work. Work can come from network functions, operating system functions, on behalf of a user, another system within the network, or from general system services, such as the clustering server jobs. Server jobs typically run in one of the basic subsystems that are shipped with the system - QSYSWRK, QSERVER, or QUSRWRK. Server jobs are most commonly associated with such functions as HTTP, Lotus Notes(R), and TCP/IP. The Work Management: Server Jobs topic in the iSeries Information Center contains more information about server jobs.

There are several reasons why configuring the server subsystems for server jobs is advantageous. Some of those reasons are as follows:

- Performance tuning is one reason to do subsystem configuration and can be done by changing routing entry information. The routing information that can be changed includes the memory pool that server jobs run in and class information. The class information is used to configure various performance attributes such as run priority, time slice, maximum threads allowed, default wait time, and more. Display Subsystem Description (DSPSBSD) command displays a menu that will give access to routing entry information. Additional performance tuning information can be found in the iSeries Information Center.

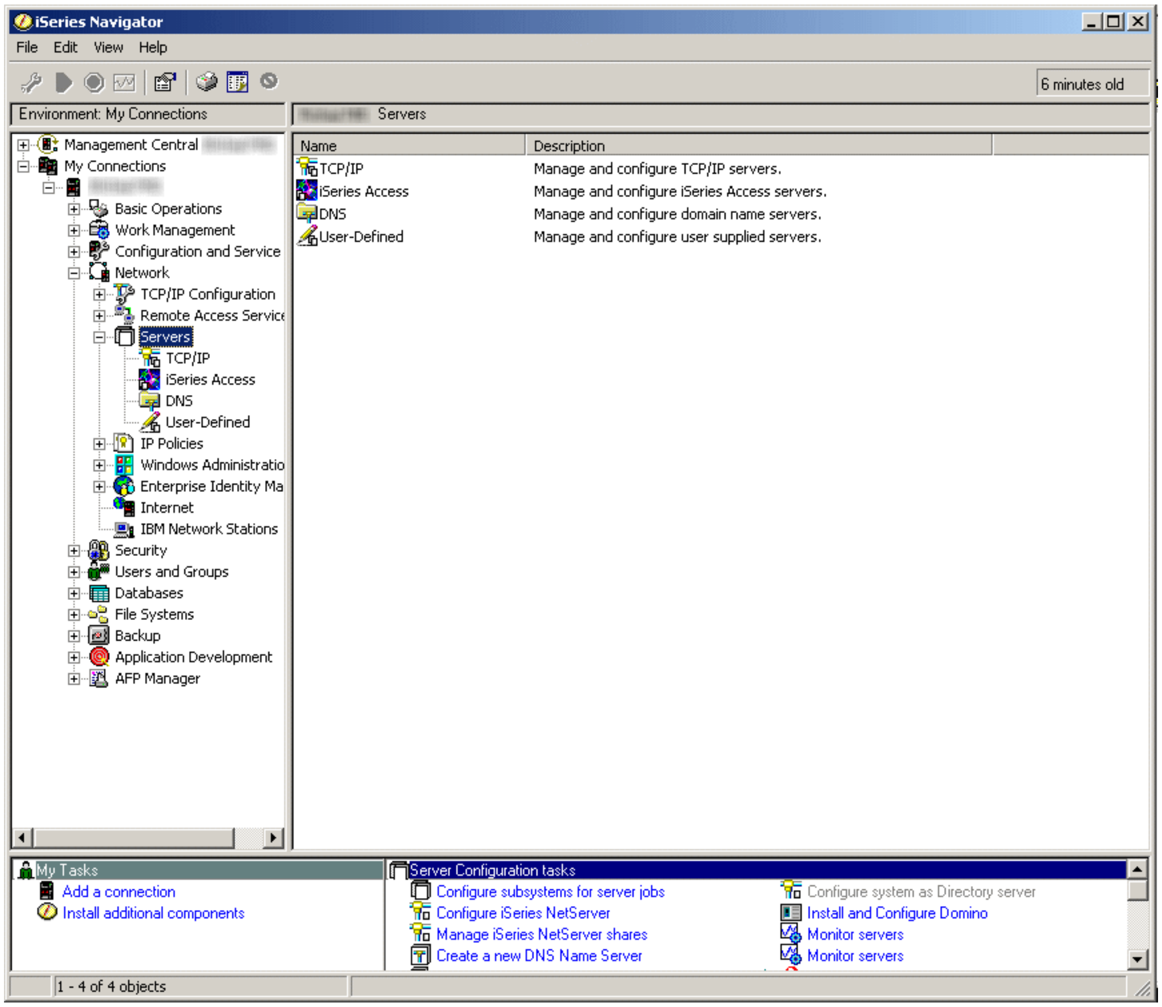
- System management is another reason to do subsystem configuration. Configuring servers to use multiple subsystems distributes system work and allows for greater control over system resources. For example, you may want to have a subsystem dedicated to database server work and let it have a large amount of system resources. Another example could be to control access to a subsystem by only allowing users on a particular subnet to send work to the subsystem. System management can also be used to aid in system maintenance. As a further example, you could have a subsystem for all employees on the east coast, a subsystem for employees in the Midwest, and another subsystem for the west coast employees. As the day progresses when the east coast employees are done working you can end the east coast subsystem without affecting the west coast and Midwest employees. By slowly taking down each subsystem as the respective employees leave, you can perform maintenance on those resources no longer in use without a large impact to users. For more information see the manage jobs or the manage subsystems information found in the Work Management topic in the iSeries Information Center.
- Spreading the work across multiple subsystems provides improved error tolerance. By distributing work across multiple subsystems, the impact of a failure in one subsystem is more contained without affecting as many potential users. A subsystem is handling over 5000 jobs at a time, if something bad happens to that subsystem where it gets into an error state; 5000 jobs are affected and could be lost. If those 5000 jobs were spread out over multiple subsystems, the result of a failure in one of these many subsystems impacts a much smaller number of jobs.

Server subsystem configuration

Some of the server jobs provided by i5/OS (R) can be configured to run in subsystems other than the default subsystem. A server can use subsystems that are shipped with the system or “User-defined server subsystems” on page 13. A server can also use multiple subsystems. Routing to the proper subsystem is controlled by IP addresses.

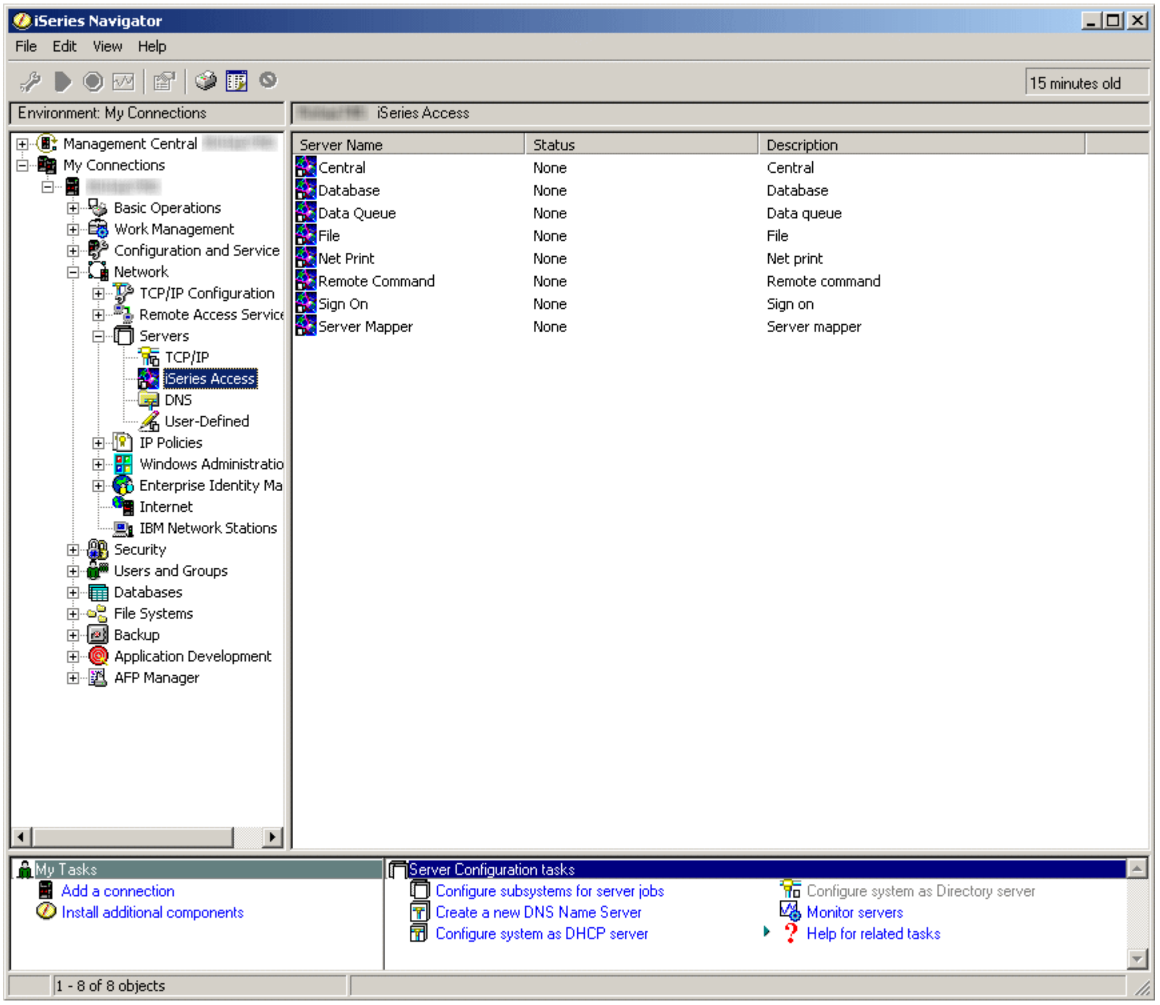
To specify which subsystem server jobs use, you must use iSeries(TM) Navigator. The following steps show how to configure server subsystems using the iSeries Navigator.

1. Open **iSeries Navigator**.
2. Expand **My Connections**.
3. Expand your **iSeries Server**.
4. Expand **Network**.
5. Expand **Servers**.



Name	Description
TCP/IP	Manage and configure TCP/IP servers.
iSeries Access	Manage and configure iSeries Access servers.
DNS	Manage and configure domain name servers.
User-Defined	Manage and configure user supplied servers.

6. Select the type of servers that you want to configure subsystems for. DDM and iSeries NetServer are **TCP/IP** servers. Central, Database, Data Queue, Files, Net Print, Remote Command, and Sign On are **iSeries Access** servers. The remaining steps use a subsystem from iSeries Access. The subsystem configuration steps are identical for each server.



Management Central [redacted]
My Connections
[redacted]
Basic Operations
Work Management
Configuration and Service
Network
TCP/IP Configuration
Remote Access Service
Servers
TCP/IP
iSeries Access
DNS
User-Defined
IP Policies
Windows Administratio
Enterprise Identity Ma
Internet
IBM Network Stations
Security
Users and Groups
Databases
File Systems
Backup
Application Development
AFP Manager

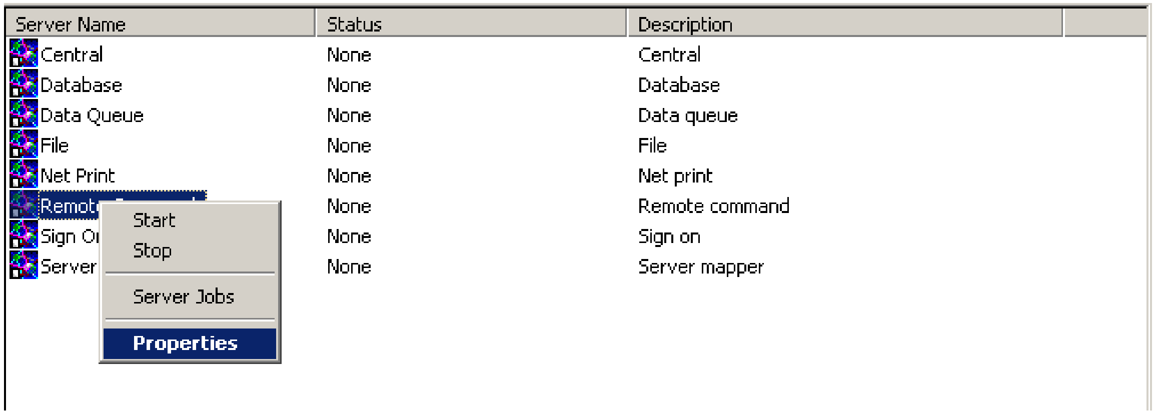
Server Name	Status	Description
Central	None	Central
Database	None	Database
Data Queue	None	Data queue
File	None	File
Net Print	None	Net print
Remote Command	None	Remote command
Sign On	None	Sign on
Server Mapper	None	Server mapper

My Tasks
Add a connection
Install additional components

Server Configuration tasks
Configure subsystems for server jobs
Create a new DNS Name Server
Configure system as DHCP server
Configure system as Directory server
Monitor servers
Help for related tasks

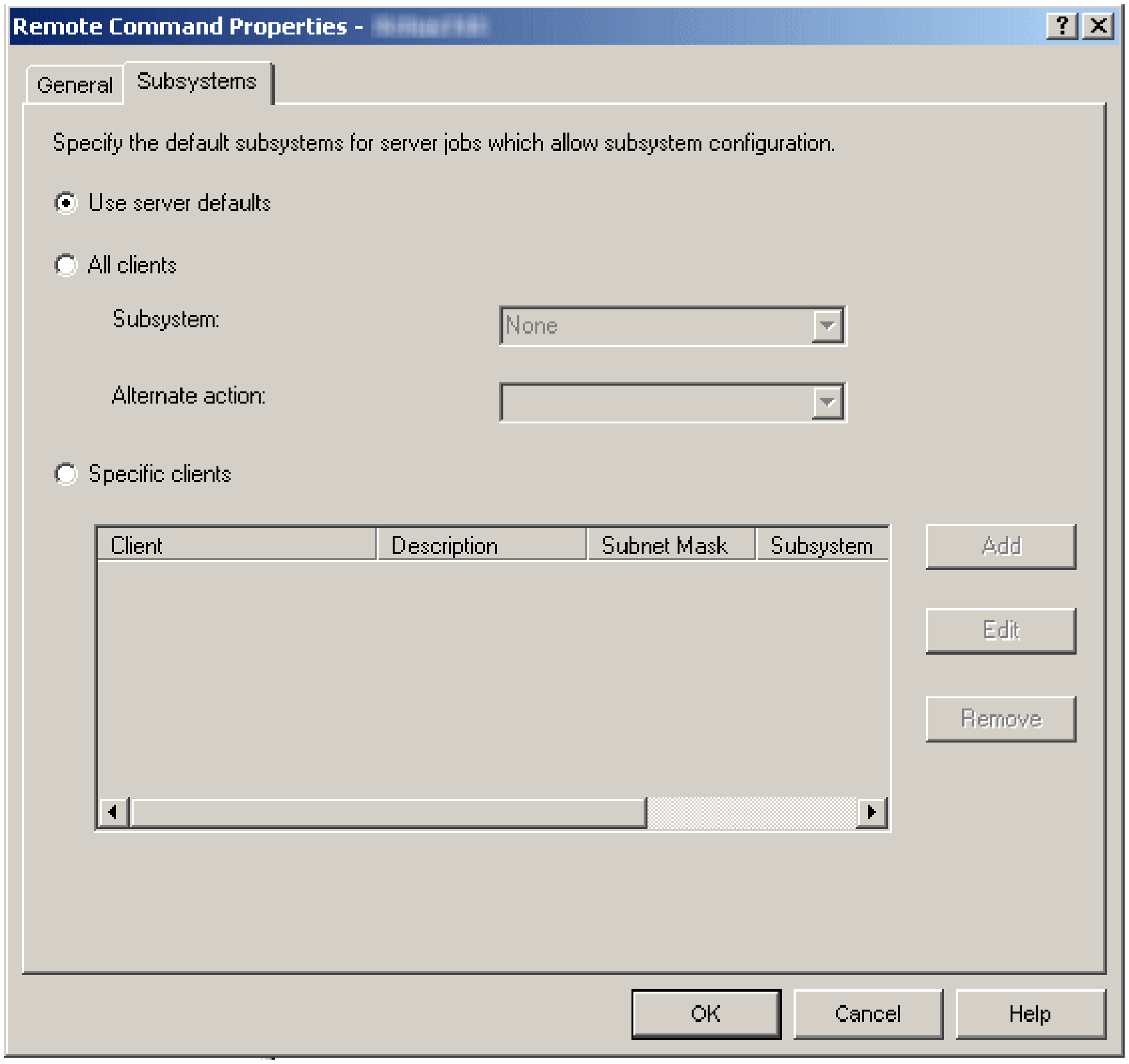
1 - 8 of 8 objects

7. In the right pane, right-click on the server that you want to configure subsystems for and select **Properties**. Another window opens, showing tabs for general and subsystem information for that server.



Server Name	Status	Description
Central	None	Central
Database	None	Database
Data Queue	None	Data queue
File	None	File
Net Print	None	Net print
Remot	None	Remote command
Sign O	None	Sign on
Server	None	Server mapper

8. Select the **Subsystems** tab.


Remote Command Properties - [redacted]

General | Subsystems

Specify the default subsystems for server jobs which allow subsystem configuration.

[✓ Use server defaults]

All clients

Subsystem: None

Alternate action:

Specific clients

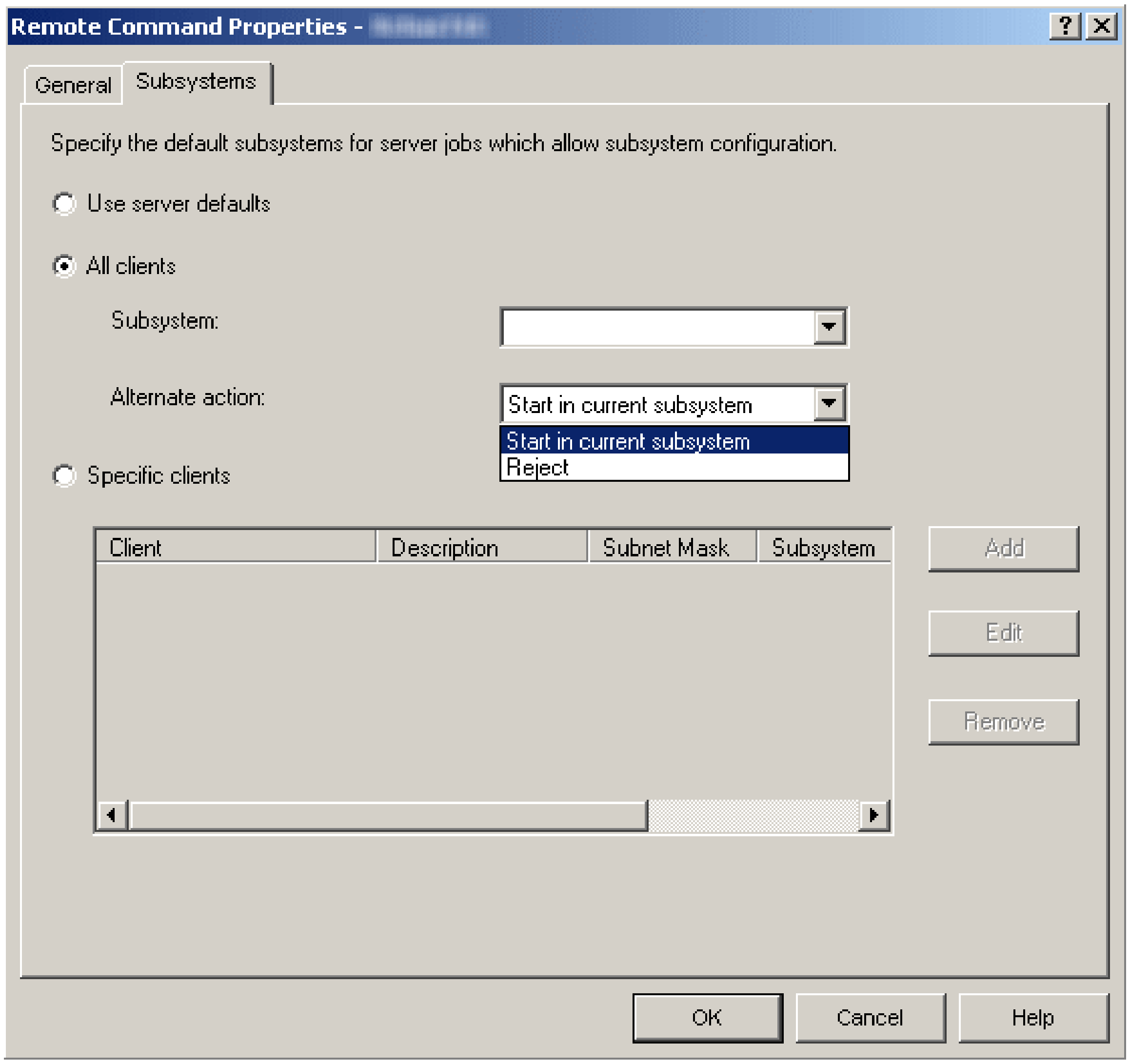
Client	Description	Subnet Mask	Subsystem

From the Subsystems tab you can specify in which subsystem you want this server's job to run. There are different ways to configure the subsystems. Select the appropriate configuration method and complete the following instructions for your selection:

- **Use server defaults**

  Select this option if you want the particular server that you are working on to use the server defaults. 'Server defaults' allow you to configure in one place the general characteristics you want for subsystems for all server jobs. You can then have individual servers uses these defaults, or you can have a more specific configuration for a specific type of server. See "Server subsystem defaults" on page 10 for more information on how to use and set the server defaults.
- **All clients**

  Select this option if all clients that use this server are to use the same subsystem and alternate action. Using 'All clients' provides an easy way to subdivide the work done by the various servers into a subsystem per type of server. For example, you can have all clients for the database server use the DATABASE subsystem, all clients for the remote command server use the RMTCMD subsystem, etc.


Remote Command Properties -

General | Subsystems

Specify the default subsystems for server jobs which allow subsystem configuration.

( ) Use server defaults

(●) All clients

Subsystem:

Alternate action: Start in current subsystem

Start in current subsystem
Reject

( ) Specific clients

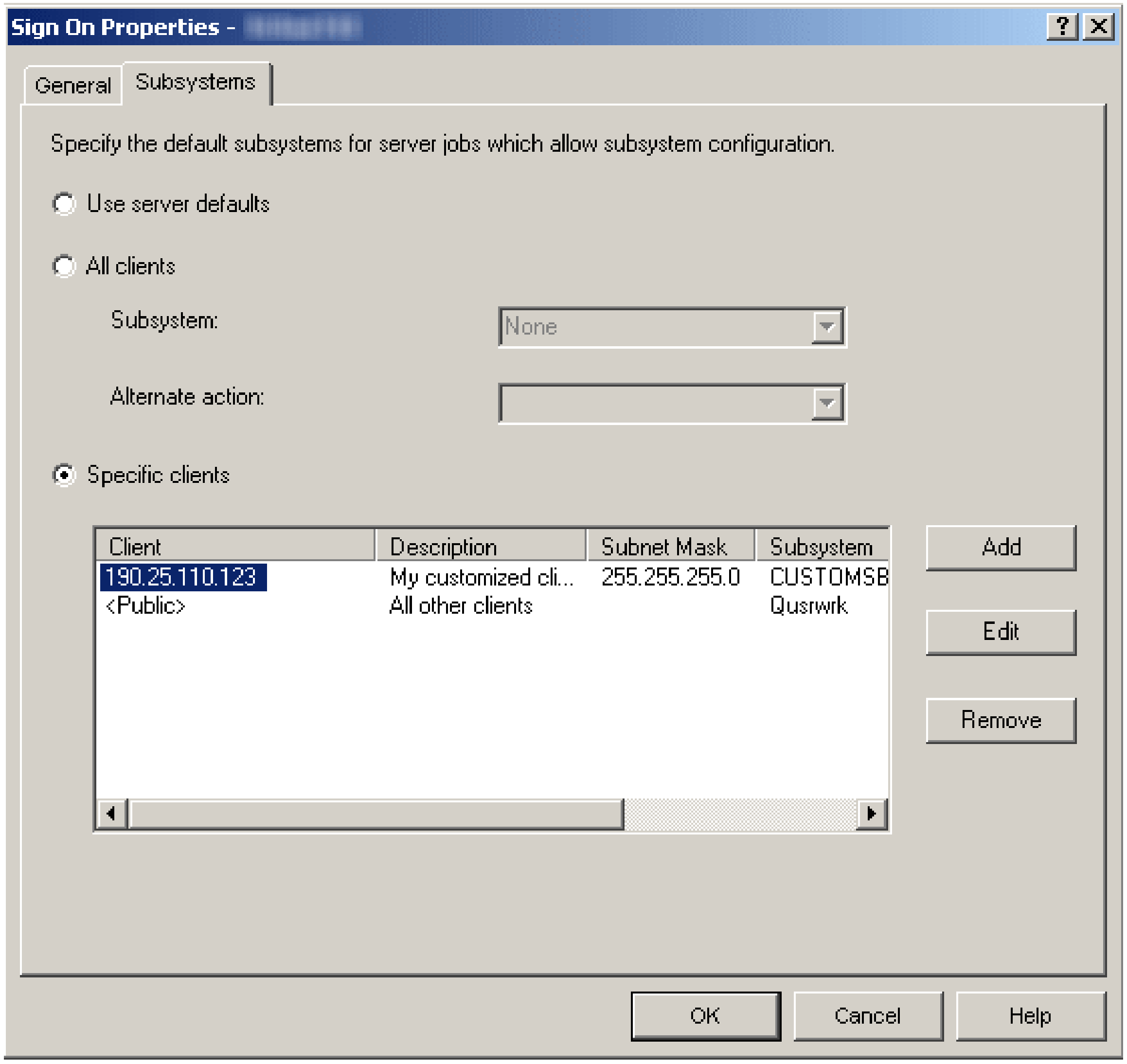
Client	Description	Subnet Mask	Subsystem

The **Subsystem** list specifies the subsystem you want this server's jobs to run in. If None is selected, the server jobs will perform the Alternate action. You can type in the ten character name of the subsystem you wish to use if it is not contained in the list. If the subsystem is not a system supplied subsystem; see the "User-defined server subsystems" on page 13 information to properly ensure the desired subsystem is setup and running correctly.

The **Alternate action** list specifies what to do if the server jobs cannot run in the specified subsystem; for example if the subsystem is not active or if you want to prohibit certain users from using specific servers. Another example where the alternative action would be taken is when the necessary prestart jobs for the given subsystem are not active. Possible values are Reject, and Start in current subsystem. When Reject is selected, the request will be rejected if it cannot run in the specified subsystem. When Start in current subsystem is selected, if the job can not run in the specified subsystem, the job will attempt to run in the same subsystem that the server daemon is running in. For the database and file servers this will typically result in the request being run in the QSERVER subsystem. For other servers this will result in requests being sent to QSYSWRK. See the server table in the iSeries Information Center for more information for server jobs, the subsystems in which they run (including server and daemon jobs), and more.

- **Specific clients**

  Select this option if you want to set up unique subsystem configuration for specific clients. When you add specific client configuration, a <Public> entry will be added to the end of the list. The <Public> entry applies to all clients not included by the specific client entries you have added.



Sign On Properties - [redacted]

General | Subsystems

Specify the default subsystems for server jobs which allow subsystem configuration.

○ Use server defaults

○ All clients

Subsystem: None

Alternate action:

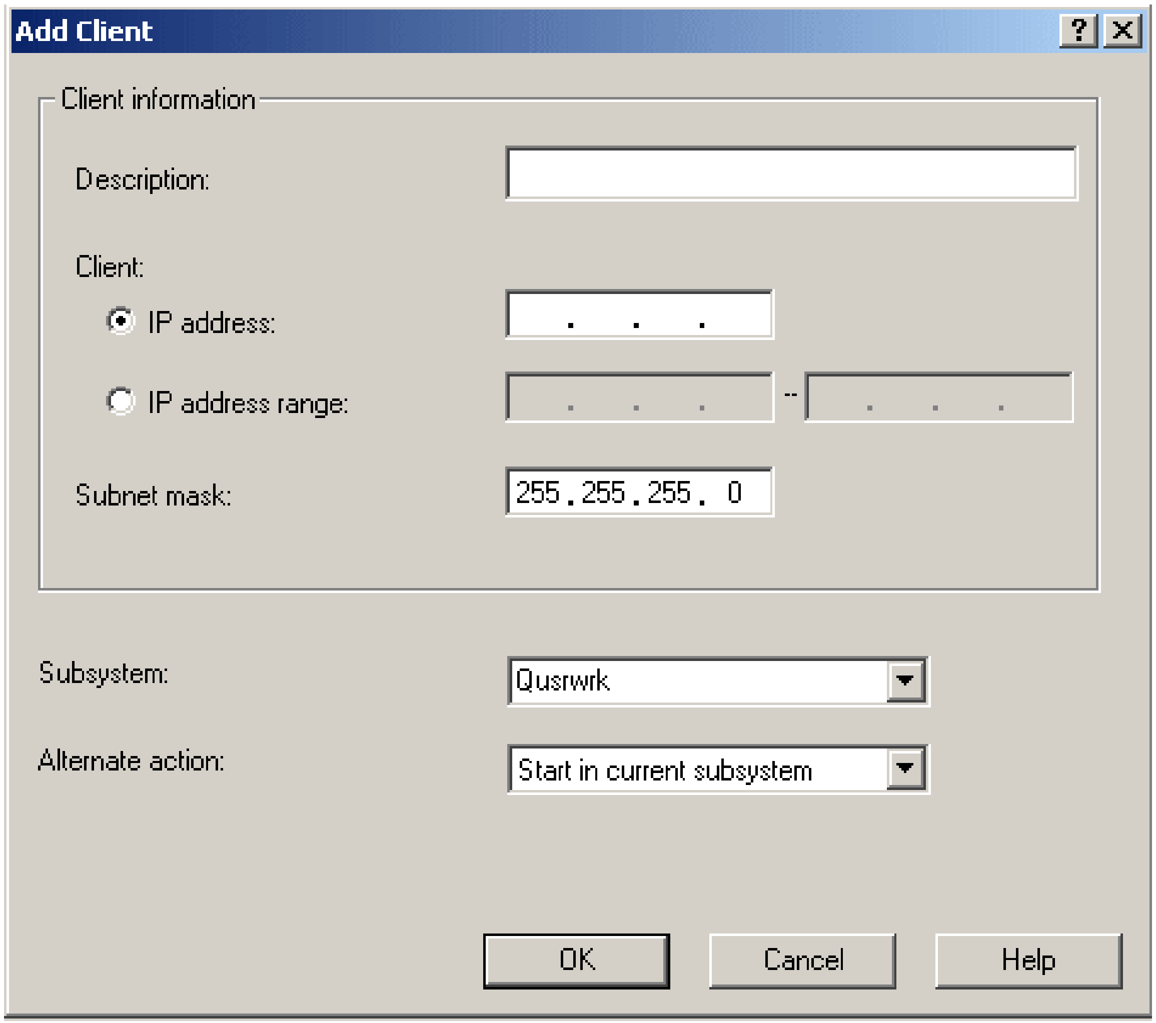
● Specific clients

Client	Description	Subnet Mask	Subsystem
190.25.110.123	My customized cli...	255.255.255.0	CUSTOMSB
<Public>	All other clients		Qusrwrk

There are different actions you can take to configure specific server subsystem clients.

– **Add**

Click Add to add a client to the list. This will take you to the Add Client dialog where you can specify the subsystem configuration for a specific client or group of clients. This will bring up a new window, the Add Client dialog.



Description:

Client:

[✓ IP address:] . . .

IP address range: . . . -- . . .

Subnet mask: 255 . 255 . 255 . 0

Subsystem: Qusrwrk

Alternate action: Start in current subsystem

From the Add Client dialog you can specify the subsystem configuration for a specific client or a group of clients.

The **Description** text box specifies a text description of the client(s) that you are configuring.

The **Client** radio button options specifies whether you use an individual IP address or IP address range. An individual IP address should be used for a single client. An IP address range is used for a group of clients. IP address ranges cannot overlap for the selected server.

The **Subnet mask** specifies the subnet mask for this IP address. The subnet mask is a unique, 32-bit integer that defines the part of the network where an interface attaches. The mask is expressed in the form xxx.xxx.xxx.xxx, where each field is the decimal representation of one byte (8 bits) of the mask. For example, the subnet mask whose hexadecimal representation is X'FFFFFF00' is expressed as 255.255.255.0.

The **Subsystem** list specifies the subsystem you want this server's jobs to run in. If None is selected, the server jobs will perform the Alternate action. You can type in the ten character name of the subsystem you wish to use if it is not contained in the list. If the subsystem is not a system supplied subsystem; see the "User-defined server subsystems" on page 13 information to properly ensure the desired subsystem is setup and running correctly.

The **Alternate action** list specifies what to do if the server jobs cannot run in the specified subsystem; for example if the subsystem is not active or if you want to prohibit certain users from using specific servers. Another example where the alternative action would be taken is when the necessary prestart jobs for the given subsystem are not active. Possible values are Reject, and Start in current subsystem. When Reject is selected, the request will be rejected if it cannot run in the specified subsystem. When Start in current subsystem is selected, if the job can not run in the specified subsystem, the job will attempt to run in the same subsystem that the server daemon is running in. For the database and file servers this will typically result in the request being run in the QSERVER subsystem. For other servers this will result in requests being sent to QSYSWRK. See the server table in the iSeries Information Center for more information for server jobs, the subsystems in which they run (including server and daemon jobs), and more.

– **Edit**

  Click **Edit** to edit a client that you have added to the list. You must select a client from the list before you can edit it. The Edit Client dialog looks identical to the Add Client dialog and has the same configuration options.

– **Remove**

  Click **Remove** to delete a client that you have added to the list. You must select the client from the list before you can remove it. You cannot remove the <Public> entry at the end of the list.

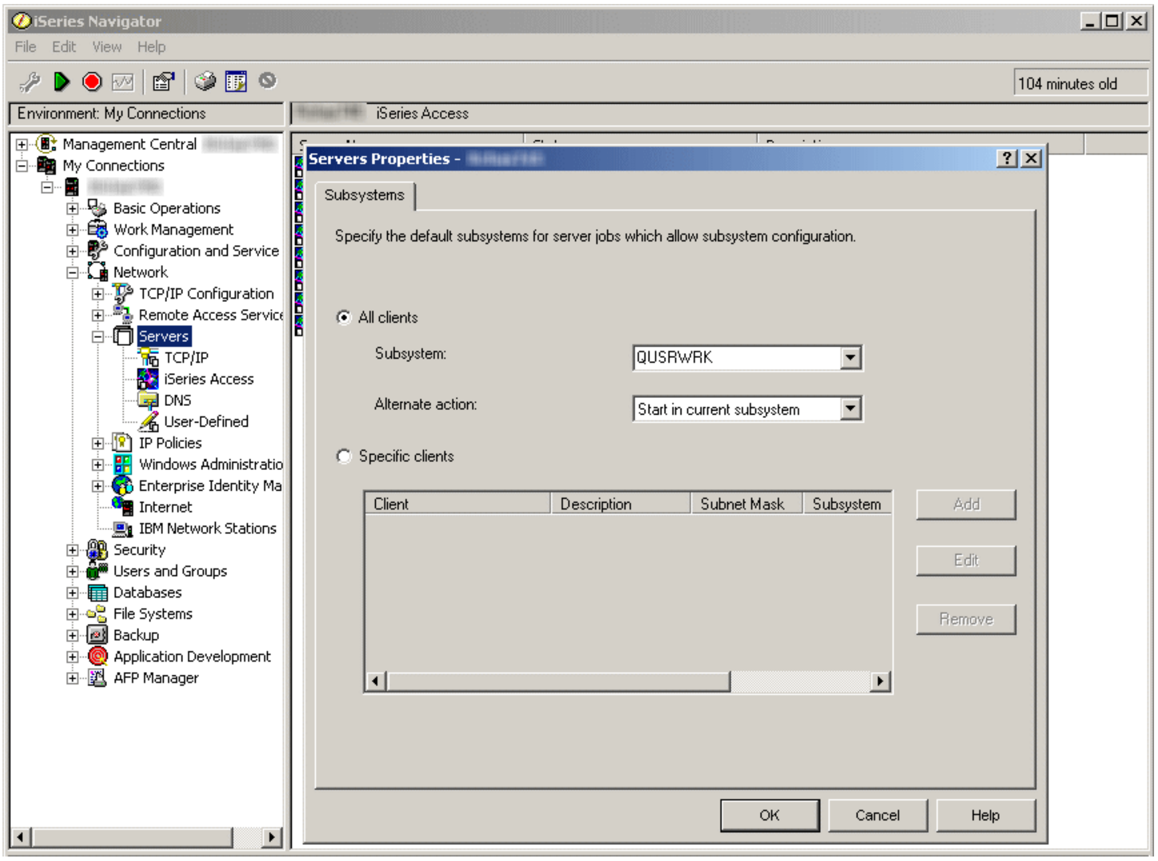
After all selections are made, click **OK** to accept and apply the specified server subsystem defaults or click **Cancel** to reject and ignore any changes made to the server subsystem default settings.

Server subsystem defaults

Setting the server defaults provides a method to configure in one place the general characteristics for subsystems for all server jobs. The individual servers can then use these defaults. To specify the server defaults, you must use iSeries$^{(TM)}$ Navigator; there is no green-screen interface for this function. The following steps show how to set the server subsystem defaults using the iSeries Navigator.

1. Open **iSeries Navigator**.
2. Expand **My Connections**.
3. Expand your **iSeries Server**.
4. Expand **Network**.
5. Select **Servers** and right-click to bring up a menu of options.
6. Open **Properties** by selecting the properties menu option.

The server defaults are specified on the Subsystem properties page for the Servers container.



Specify the default subsystems for server jobs which allow subsystem configuration.

[● All clients]

Subsystem: QUSRWRK

Alternate action: Start in current subsystem

○ Specific clients

Client	Description	Subnet Mask	Subsystem

From the subsystem property page you can specify which default subsystems you want all server jobs to run in. There are different ways to configure the subsystems. Select the appropriate configuration method and complete the following instructions for your selection:

- **All clients**

  Specifies that all clients for all servers are to use the same subsystem and alternate action.

  The **Subsystem** list specifies the subsystem you want this server's jobs to run in. If None is selected, the server jobs will perform the Alternate action. You can type in the ten character name of the subsystem you wish to use if it is not contained in the list. If the subsystem is not a system supplied subsystem; see the "User-defined server subsystems" on page 13 information to properly ensure the desired subsystem is setup and running correctly.

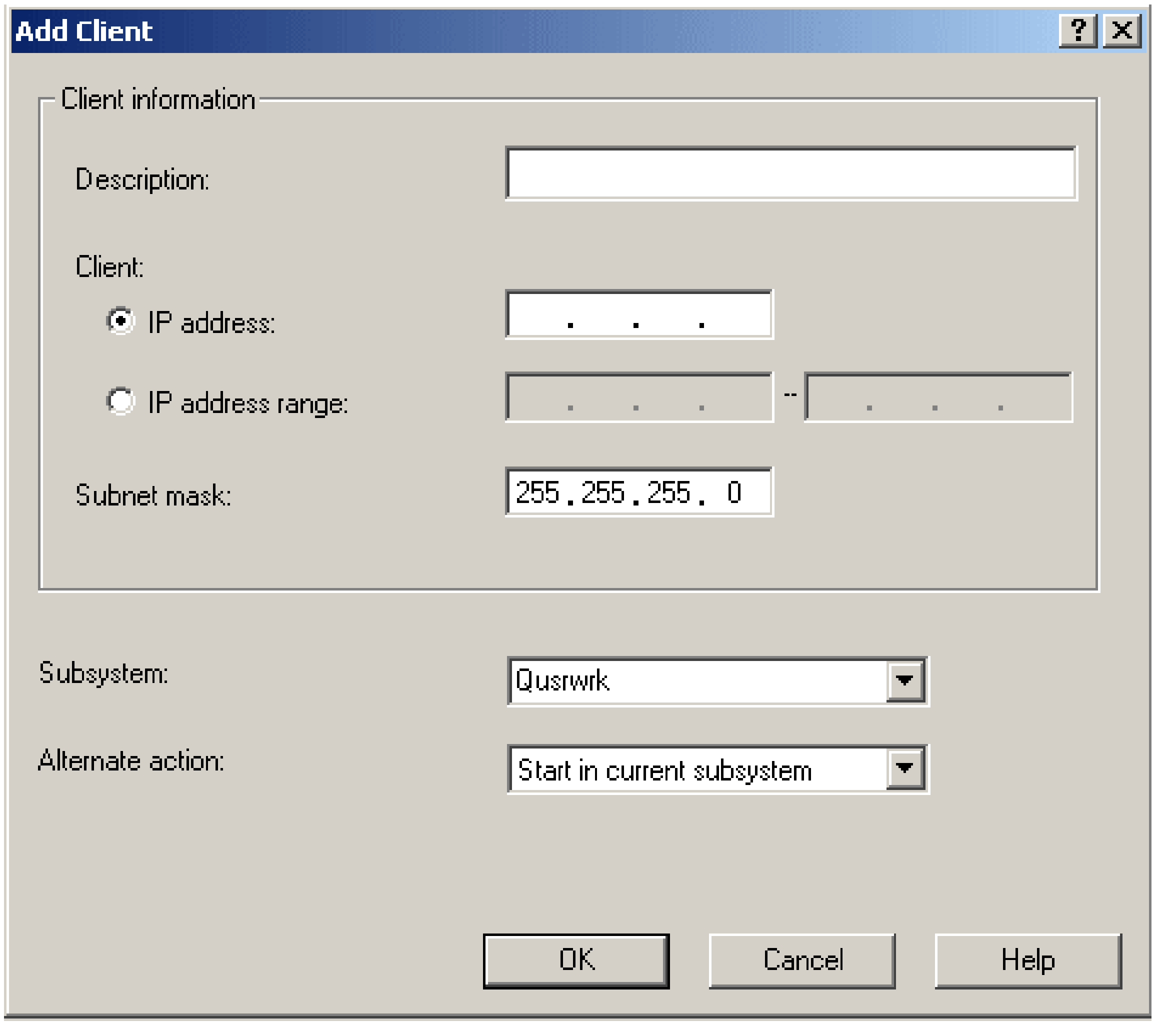
  The **Alternate action** list specifies what to do if the server jobs cannot run in the specified subsystem; for example if the subsystem is not active or if you want to prohibit certain users from using specific servers. Another example where the alternative action would be taken is when the necessary prestart jobs for the given subsystem are not active. Possible values are Reject, and Start in current subsystem. When Reject is selected, the request will be rejected if it cannot run in the specified subsystem. When Start in current subsystem is selected, if the job can not run in the specified subsystem, the job will attempt to run in the same subsystem that the server daemon is running in. For the database and file servers this will typically result in the request being run in the QSERVER subsystem. For other servers this will result in requests being sent to QSYSWRK. See the server table in the iSeries Information Center for more information for server jobs, the subsystems in which they run (including server and daemon jobs), and more.

- **Specific clients**

  Specifies that you want to set up unique subsystem configuration for specific clients. When you add specific client configuration, a <Public> entry will be added to the end of the list. The <Public> entry applies to all clients not included by the specific client entries you have added.

  There are different actions you can take to configure specific default server subsystem clients.

– **Add**
Click Add to add a client to the list. This will take you to the Add Client dialog where you can specify the subsystem configuration for a specific client or group of clients. This will bring up a new window, the Add Client dialog.



Subnet mask: 255.255.255.0
Subsystem: Qusrwrk
Alternate action: Start in current subsystem

From the Add Client dialog you can specify the subsystem configuration for a specific client or a group of clients.

The **Description** text box specifies a text description of the client(s) that you are configuring.

The **Client** radio button options specify whether you use an individual IP address or IP address range. An individual IP address should be used for a single client. An IP address range is used for a group of clients. IP address ranges cannot overlap for the selected server.

The **Subnet mask** specifies the subnet mask for this IP address. The subnet mask is a unique, 32-bit integer that defines the part of the network where an interface attaches. The mask is expressed in the form xxx.xxx.xxx.xxx, where each field is the decimal representation of one byte (8 bits) of the mask. For example, the subnet mask whose hexadecimal representation is X'FFFFFF00' is expressed as 255.255.255.0.

The **Subsystem** list specifies the subsystem you want this server's jobs to run in. If None is selected, the server jobs will perform the Alternate action. You can type in the ten character name of the subsystem you wish to use if it is not contained in the list. If the subsystem is not a system supplied subsystem; see the "User-defined server subsystems" on page 13 information to properly ensure the desired subsystem is setup and running correctly.

The **Alternate action** list specifies what to do if the server jobs cannot run in the specified subsystem; for example if the subsystem is not active or if you want to prohibit certain users from using specific servers. Another example where the alternative action would be taken is when the necessary prestart jobs for the given subsystem are not active. Possible values are Reject, and Start in current subsystem. When Reject is selected, the request will be rejected if it cannot run in the specified subsystem. When Start in current subsystem is selected, if the job can not run in the specified subsystem, the job will attempt to run in the same subsystem that the server daemon is running in. For the database and file servers this will typically result in the request being run in the QSERVER subsystem. For other servers this will result in requests being sent to QSYSWRK. See the server table in the iSeries Information Center for more information for server jobs, the subsystems in which they run (including server and daemon jobs), and more.

– **Edit**

  Click **Edit** to edit a client that you have added to the list. You must select a client from the list before you can edit it. The Edit Client dialog looks identical to the Add Client dialog and has the same configuration options.

– **Remove**

  Click **Remove** to delete a client that you have added to the list. You must select the client from the list before you can remove it. You cannot remove the <Public> entry at the end of the list.

After all selections are made, click **OK** to accept and apply the specified server subsystem defaults or click **Cancel** to reject and ignore any changes made to the server subsystem default settings.

User-defined server subsystems

The following article describes how to create a new user-defined server subsystem that is based off of the QUSRWRK system supplied subsystem. User-defined subsystems can be used for “Server subsystem configuration” on page 3.

One of the first planning steps is to determine the naming convention you will use for your subsystems. The naming convention can be something simple, something more reflective of your users (INVENTORY, PGMR, ORDERENT), or reflective of the geographic location in which your users reside (EAST, CENTRAL, WEST). Decide upon a naming convention that will be meaningful for your system administration.

The steps below are described as if the subsystem was created interactively. You can use a CL program to create your subsystems so you can easily recreate your configurations for recovery purposes.

1. Create a library to store your subsystem configuration objects in. In this example, we will use SBSLIB.

   ```
   CRTLIB SBSLIB TEXT('Library to hold subsystem configuration objects')
   ```

2. Create a class. The class defines certain performance characteristics for your subsystem including run priority, time slice, and default wait times.

   The following example creates a class MYSBS in the SBSLIB library that has a priority of 20, a time slice of 2000 milliseconds, and has a default wait time of 30 seconds.

   ```
   CRTCLS SBSLIB/MYSBS RUNPTY(20) TIMESLICE(2000) DFTWAIT(30)
   TEXT('Custom Subsystem Class')
   ```

   For more information on creating a class see the Create Class (CRTCLS) command description.

   Note:

   You can create a single class to use for all of your subsystems, or you can create a class for each subsystem. Which you choose depends upon whether you want to customize some of the performance settings for particular subsystems. IBM(R)-supplied subsystems are shipped with a class created for each subsystem, and the name of the class is the same as the name of the subsystem. If you do not create a class for each subsystem with the same name as the subsystem, you need to specify the class name on the Add Routing Entry (ADDRTGE) command since the default for the class (CLS) parameter is *SBSD, which means the class name has the same name as the subsystem description.

The following example demonstrates why you may want different classes for different subsystems. If you want to have your critical users have a higher run priority than the remainder of your users, you could accomplish this by having those critical users run in their own subsystem, and the class used for that subsystem could specify a higher run priority. Remember for run priority that a lower number gives a higher priority.

3. Create the subsystem description. You will repeat this step for each subsystem you need to define. More information on how to create a subsystem description can be found in the iSeries Information Center.

   The following example creates a subsystem description with attributes identical to those of QUSRWRK. Job queue, routing entries, and prestart job information will have to be added manually at a later time.

   ```
   CRTSBSD SBSD(SBSLIB/MYSBS) POOLS((1 *BASE)) TEXT('Custom Server Subsystem')
   ```

   After a subsystem description is created, use the Display Subsystem Description (DSPSBSD) command to display the subsystem description information.

   As an alternative to creating a subsystem description from scratch, you can copy an existing subsystem description using the Create Duplicate Object (CRTDUPOBJ) command. The following example creates a subsystem description with attributes identical to those of QUSRWRK. This will also copy job queue, routing entry, and prestart job information form QUSRWRK. Depending on the requirements for your subsystem, these values may have to be changed.

   ```
          CRTDUPOBJ OBJ(QUSRWRK) FROMLIB(QSYS) OBJTYPE(*SBSD) TOLIB(SBSLIB)
   NEWOBJ(MYSBS)
   ```

   This report assumes that a subsystem description was created from scratch

   Note:

   The storage pools used are specified on the subsystem description. If you want to isolate a set of users to a pool of their own, you can do this by specifying a pool ID with a storage size and activity level specifically for the subsystem, rather than using the *BASE pool. You could also define a shared pool with the Change Shared Storage Pool (CHGSHRPOOL) command and specify that shared pool on the subsystem description.

   The subsystem description also defines the number of jobs allowed to run in the subsystem. The following parameters are available the Create Subsystem Description (CRTSBSD) command, but add complexity that may make it much easier to unintentionally prevent users from accessing the subsystem, thus the recommendation to not use these parameters.

   Activity Level (MAXACT): The activity level determines the maximum number of threads that can be actively running at one time. More threads than this can be active within the subsystem, but the activity level determines the number that can be actively running at any given point in time.

   Maximum Jobs (MAXJOBS): The maximum number of jobs parameter determines the number of jobs running in the subsystem. This value cannot be exceeded. Any attempt to start additional jobs when the maximum number of jobs is already running will fail.

4. Create a job queue for the subsystem, using the same name as the subsystem name and add a job queue entry to the subsystem description. In general a job queue for the subsystem will be required. Having a job queue allows server jobs to be submitted to the subsystem when needed. Server jobs on the job queue will then be processed. Also if you will be using the Transfer Job (TFRJOB) command to transfer jobs into your custom subsystem, you will want a job queue.

   ```
   CRTJOBQ JOBQ(SBSLIB/MYSBS)
   ```

   ```
   ADDJOBQE SBSD(SBSLIB/MYSBS) JOBQ(SBSLIB/MYSBS) MAXACT(*NOMAX)
   ```

   For more information on each command see the Create Job Queue (CRTJOBQ) command description and the Add Job Queue Entry (ADDJOBQE) . command description. The iSeries Information Center also has information on how a job queue works .

5. Add a routing entry to the subsystem.

The following is an example of adding a routing entry that is designed to be a generic entry to catch all requests that do not match a specific routing entry. The routing entries added to any user-defined subsystem can vary from this example and will depend on the specific requirements for the subsystem. A good basis for setting up routing entries on a user-defined subsystem is to make them identical to the routing entries used on QUSRWRK.

```
       ADDRTGE SBSD(SBSLIB/MYSBS) SEQNBR(9999) CMPVAL(*ANY) PGM(QSYS/QCMD)
CLS(SBSLIB/MYSBS)
```

If you look at the routing entries shipped on the system for QUSRWRK, you will see there are several additional routing entries. If you need those functions, add those routing entries to your customized subsystem descriptions as well. It is recommended to copy the routing entries from QUSRWRK into the new user-defined subsystem.

See the Add Routing Entry (ADDRTGE) command description for more information.

Note:

Maximum Activity Level (MAXACT): The maximum activity level refers to the maximum number of routing steps that can be active through this routing entry. It is highly recommend to use the default value of *NOMAX.

The ADDRTGE command specifies which pool identifier to use and the default on the command is 1. If you had set up your subsystem description with dedicated pools, be sure to specify the appropriate pool identifier on the routing entry.

If the name of your class is different from the name of your subsystem description, you will need to specify the class on the ADDDRTGE command.

You can get quite complex with routing entries. If you need to investigate the options available refer to the Work Management topic in the Information Center.

6. Add prestart job entries to the subsystem. A prestart job is a job that is started and waits for work to be dispatched to it. It is particularly useful for subsystems to have a number of prestart jobs available to handle requests that are submitted to the subsystem.

   The following is an example of adding a prestart job entry. The prestart job entries added to any user-defined subsystem can vary from this example and will depend on the specific requirements for the subsystem. It is recommended that the prestart job entries on a user-defined subsystem be identical to the prestart job entries used on QUSRWRK.

```
ADDPJE SBSD(SBSLIB/MYSBS) PGM(QSYS/QZSOSIGN) INLJOBS(50) THRESHOLD(4)
JOB(QZSOSIGN) JOBD(QSYS/QZBSJOBD) CLS(QGPL/QCASERVR) STRJOBS(*YES)
```

   Do not use the defaults on the command. For information on balancing the initial number of jobs, threshold, additional number of jobs parameters and other important prestart job settings, see the tuning prestart job entries information. See the Add Prestart Job Entry (ADDPJE) command description for more information on how to add a prestart job entry to a subsystem description.

Note:

By initializing the job before work arrives, the overhead and time of doing initialization is avoided, allowing for a much higher throughput of work. It is very important that the prestart job entries be configured correctly to be able to handle the expected amount of work and the expected amount of requests sent to them. If not done correctly, jobs may end up taking an alternative action or a degradation in performance will be seen as additional prestart jobs are created if the current pool of prestart jobs is exhausted. See “Server subsystem configuration” on page 3 information for details on alternative actions for subsystems.

The following parameters are used to determine the number of prestart jobs to initially run and what to do when the current pool of initial prestart jobs is in use.

- Initial number of jobs (INLJOBS): Specifies the initial number of prestart jobs that are started when the subsystem specified on the Subsystem description (SBSD) parameter is started. The value of this parameter must be less than or equal to the value of the Maximum number of jobs (MAXJOBS) parameter. The value of this parameter must be greater than or equal to the value of the Threshold (THRESHOLD) parameter. The current workload is not the same as the initial number of jobs. A good rule of thumb is to set the initial number of jobs to the expected workload plus the threshold.
- Threshold (THRESHOLD): Specifies when additional prestart jobs are started. When the pool of available jobs (jobs available to service requests) is reduced below this number, more jobs (specified on the Additional number of jobs (ADLJOBS) parameter) are started and added to the available pool. The value of this parameter must be less than or equal to the value specified on the Initial number of jobs (INLJOBS) parameter. The threshold should reflect the system workload and the amount of new work coming in once all available prestart jobs unavailable.
- Additional number of jobs (ADLJOBS): Specifies the additional number of prestart jobs that are started when the number of prestart jobs drops below the value specified on the Threshold (THRESHOLD) parameter. The value specified on this parameter must be less than the value specified on the Maximum number of jobs (MAXJOBS) parameter.

For more information on prestart jobs, see the prestart jobs articles in the Work Management section of the iSeries Information Center.

Once the subsystem descriptions have been created, use the Display Subsystem Description (DSPSBSD) command to display the various attributes of the subsystem and verify that the set up has been done correctly.

Start the subsystems using the Start Subsystem (STRSBS) command. To verify that work is being processed by the subsystem, “Server subsystem configuration” on page 3 to use a user-defined subsystem and make sure appropriate clients are up and running in the subsystem. The Work with Subsystem Jobs (WRKSBSJOB) command can help verify that jobs are running in the user-defined subsystem.

Any user-defined subsystem will have to be started before any servers that use the user-defined subsystem are started. The best way to ensure that a user-defined subsystem is available for a server, is to start the subsystem before TCP/IP starts. There are two ways to do this. This first is changing the IPL start-up program . This has the benefit of automatically starting subsystems during an IPL, which avoids having to manually start them each time the system is restarted or started from the restricted state. A second way to control subsystem start-up is to change the STRTCP IPL attribute to *NO and then to manage starting the subsystems, TCP/IP, and server jobs in the right order in the system startup program. See the Start TCP/IP (STRTCP) command description for more information including the IPL autostart information. There is more information on how to start the server in the iSeries Information Center.

References and resources

iSeries$^{(TM)}$ Information Center

- Work Management
- System Values

Disclaimer

Information is provided ″AS IS″ without warranty of any kind. Mention or reference to non-IBM products is for informational purposes only and does not constitute an endorsement of such products by IBM.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.

Printed in USA