Documentation Library

Administrator Guide

WELCOME TO AXES **4**
ROAD MAP 4
ABOUT AXES 5
aXes Terminal Server *5*
aXes Spool File Server *5*
aXes Data Explorer Server *6*
aXes Application Server *7*
SECTION 1 - AXES LICENSING **8**
PROCESSOR AND SESSION LICENSES 8
LICENSE EXPIRY AND RENEWAL 8
TROUBLESHOOTING LICENSES 9
SECTION 2 - MANAGING AXES SECURITY **10**
SERVER SECURITY 10
aXes Application Server *10*
aXes Data Explorer Server *10*
aXes Spool File Server *10*
aXes Terminal Server *10*
BASIC AUTHENTICATION SCHEME 11
AXES ACCESS CONTROL LISTS 11
Default aXes Access Control Lists *11*
aXes Access Control List Commands *11*
Creating aXes Access Control Lists *12*
Deleting aXes Access Control Lists *12*
SECTION 3 - MANAGING AXES USERS **13**
USER COMMANDS 13
ADDING USERS TO AN AXES ACCESS CONTROL LIST 13
CHANGING USER PASSWORDS 14
REMOVING USERS FROM AN AXES ACCESS CONTROL LIST 14
WORKING WITH AXES USERS 14
SECTION 4 - AUTHORITIES AND PERMISSIONS **15**
USER PROFILES 15
Adopted Authority and System State *16*
Programs that adopt authority 16
Programs that run in system state 16
aXes programs you can alter or delete 17
Remove adopt authority for these aXes programs 18
Do not remove adopt authority for these aXes programs 18
aXes programs not to delete 18
Exclude authority *18*
UNIX-STYLE PERMISSIONS 19
Overview *19*
Users and Access *19*
Files and Permissions *20*
Directories and Permissions *20*
How Permissions are Calculated *20*
How Permissions are Granted *20*
AXES APPLICATION SERVER 21
AXES DATA EXPLORER SERVER 21
AXES SPOOL FILE SERVER 21
AXES TERMINAL SERVER 22

Content © LANSA Group
All Rights Reserved

aXes Gateway Interface 22
Common Gateway Interface 22
FastCGI Servers 22

SECTION 5 - UNDERSTANDING FASTCGI 23

About the Common Gateway Interface (CGI) 23
aXes FastCGI Servers 23
Operating aXes Application Server 23
Configuring FastCGI External Server 24

SECTION 6 - MANAGING AXES SERVERS 26

Defining aXes Server Instances 26
aXes Terminal Server 27
Internal server configuration 27
External server configuration 27
aXes Data Explorer Server 28
Internal server configuration 28
External server configuration 28
aXes Spool File Server 28
Internal server configuration 28
External server configuration 29
Displaying aXes Server Instances 29
Starting and Stopping aXes Servers 29
Starting the Servers 29
Stopping the Servers 30
When to Stop and Restart aXes Servers 30
Starting aXes Servers when the System Starts 31
Working with aXes Server Instances 31

SECTION 7 - BACKUP AND RECOVERY 32

About Backing up aXes 32
Backup Tips 32
Libraries and Directories for aXes 32
Backing up aXes Programs and Product Files 33
Backing up Changed Objects from aXes 33
Disaster/Recovery for aXes Components 34
aXes Libraries 34
aXes Directories 34
Changed Objects 34

APPENDIX A – BROWSER SETTINGS 35

Cache Control 35
Advanced Settings 35
Browsing 35
Use HTTP 1.1 protocol 36
Security 36
Proxy Servers 37

Content © LANSA Group
All Rights Reserved

Welcome to aXes

This document describes aXes and explains how to use the product. For the latest information about aXes and support, refer to the aXes web site (axeslive.com).

Road Map

Use the following table to find information that will help you understand concepts, how to use aXes, and manage users, servers and security.

If you want to ...	Refer to ...
Understand licensing	Section 1. aXes Licensing
Manage security	Section 2. Managing aXes Security
Manage users	Section 3. Managing aXes Users
Learn about authorities	Section 4. Authorities and Permissions
Understand FastCGI	Section 5. Understanding FastCGI
Manage servers	Section 6. Managing aXes Servers
Learn about recovery	Section 7. Backup and Recovery

Topics available in other guides are:

If you want to ...	Refer to ...
Apply maintenance to aXes	aXes Product Suite Installation Guide
Configure aXes Application Server	aXes Reference
Configure Data Explorer Server	aXes Reference
Configure Spool File Server	aXes Reference
Configure Terminal Server	aXes Reference
Create keyboard maps	aXes Reference
Install aXes	aXes Product Suite Installation Guide
Learn about scenarios	aXes Reference
Learn about system requirements	aXes Product Suite Installation Guide
Remove aXes	aXes Product Suite Installation Guide
Set up Secure Sockets Layer	aXes Reference
Solve problems	aXes Reference
Tune aXes Application Server	aXes Reference
Tune Terminal Server	aXes Reference
Understand aXes commands	aXes Reference

Content © LANSA Group
All Rights Reserved

For detailed information about using aXes, including installation, configuration and command syntax diagrams, see the aXes Product Suite Installation and aXes Reference guides.

About aXes

aXes comprises an application server, a terminal server, a spool file server, a data explorer server, and a set of tools that allow you to access existing IBM i (System i, iSeries and AS/400) 5250 applications via a TCP/IP network using a web browser.

The terminal server components are called aXes-TS and aXes-TS2

The spool file server components are called aXes-WS and aXes-WS2

The data explorer server components are called aXes-DE and aXes-DE2

The application server component is called aXes-W3.

aXes-TS2, aXes-WS2 and aXes-DE2 are enhanced versions of aXes-TS, aXes-WS and aXes-DE.

Multiple instances of each server can be active at the same time. CL commands are provided to start, stop and manage these servers.

aXes Terminal Server

aXes Terminal Server provides access to IBM i terminal sessions from a web server. It offers all of the advanced features found in desktop-based solutions within a browser-based interface. aXes Terminal Server reduces your total cost of ownership (TOC), for accessing terminal sessions, by exploiting the infrastructure of the Web and Internet-based protocols. aXes Terminal Server provides unprecedented control over server connectivity combined with centralized configuration, licence management, and security.

The aXes Terminal Server is an advanced solution that provides access to enterprise applications via an industry standard Web browser interface in XML format over the Web or company intranet. Using a browser, users simply access an URL — eliminating the need to install software and user configurations on the individual desktops. Users access their server-based applications in a variety of styles, with a click of the mouse via their web browser. The aXes Terminal Server uses open standards to ensure access between a browser and applications running on a server is managed quickly, efficiently, and securely, resulting in lower network traffic than native terminal protocols. You get reduced telecommunication, administration, and client management costs with decreased complexity.

Terminal sessions created by the aXes Terminal Server can optionally access the full batch CPW capacity of your server. These terminal sessions have the appearance and behaviour of normal interactive jobs but they are not counted as interactive jobs by the interactive CPW monitor. Terminal sessions created by the aXes Terminal Server are independent of terminal sessions created by other mechanisms such as Telnet. Telnet and other emulator sessions continue to be counted against available interactive CPW.

The ability to access the full batch CPW is enabled by the aXes Intersession configuration directive. See the Terminal Server configuration section in the aXes Reference for a detailed explanation of the syntax.

aXes Spool File Server

aXes Spool File Server provides convenient, high speed access to IBM i output queues and spooled file data using the HTTP protocol. It allows you to manage output queues and spooled files using your browser.

Spooled content can be displayed in a variety of user selectable and defined formats:

Deliver spooled content in a variety of formats via your browser
Spooled files can be manipulated with commands run directly from the browser

Content © LANSA Group
All Rights Reserved

Spool file output can be integrated with desktop applications (including email and Microsoft Excel)
Management functions are provided for output queues and spooled content
Server-side script can be tailored (e.g. allow for page range selection of print, insert corresponding pictures based on product codes etc.)

Print-ready documents are delivered automatically in the following formats:

HTML	Format spooled data into customisable web-ready documents
PDF	Deliver print-ready PDF files
Text	Very fast access for large spooled files
XML	Publish spooled data using XSL transformations for speed and flexibility

aXes Spool File Server requires no client configuration and deployment.

aXes Spool File Server integrates with aXes Terminal Server when the user has an established a terminal session with a server. Conversion of spooled data to XML, combined with event driven dispatching provides a very fast and effective e-business delivery mechanism.

aXes Data Explorer Server

aXes Data Explorer is supplied as an additional feature of aXes Application Server. It provides convenient, high-speed access to DB2/400 and other databases using the HTTP protocol and a web browser.

Access databases from a browser:

Easy to use, you can create your first data query in minutes
Create distributed SQL database queries for any accessible data source
Support any browser for query definition and web publication
Query output can be easily integrated with desktop business applications such as Word, Excel etc.
Launch saved queries from any web page with a standard URL reference.
Zero-client configuration and deployment, server side script can be tailored.
SQL knowledge and client side configuration are not required.
Improve data accessibility from a browser.

Web ready database content is delivered for a wide variety of everyday business uses:

Data transfer	Query results can be saved or printed locally on your PC or anywhere in a network
Web publication	Deliver real-time, web-ready query results when and where they are needed.
Desktop integration	Query results can be integrated with desktop applications like Microsoft Excel.

Content © LANSA Group
All Rights Reserved

aXes Data Explorer Server requires no client configuration and deployment.

aXes Application Server

aXes Application Server is a fast and secure HTTP Server for the IBM i with integrated website acceleration technology. Its features include:

Logging
SSL support
Authentication
HTTP 1.1 support
Multiple instances
Embedded scripting
Embedded database support
Simple installation and configuration
A light foot print; small memory utilization and low CPU consumption

The aXes Application Server also provides tools for workload and user management.

Content © LANSA Group
All Rights Reserved

Section 1 - aXes Licensing

This section explains the different licensing methods used by aXes and helps you resolve problems related to licensing.

Processor and Session Licenses

aXes is licensed by processor or session. Some components are available only with processor licensing while others are available in your choice of processor or session licensing.

See the installation planning section in the aXes Product Suite Installation Guide for a list of components and the available licensing options.

Processor usage and session usage is counted across Logical Partitions. This means you must purchase enough processor licences and session licences to cover your total usage requirements. You may install aXes into multiple partitions as long as the total usage does not exceed the licensed usage you have purchased.

aXes is a licensed program product. It will function without a license key for a trial period (for example 30 days) from the date of first use. The trial period is intended only for evaluation purposes. The product will cease to operate after the trial period expires without an installed licence key for each component.

License Expiry and Renewal

The aXes licence term is by version and release (i.e., VxRy). You will need a new licence key if the term of the licence changes. For example, aXes versions 120 to 123 will operate on the same licence keys because the licence term is V1R2. However, aXes version 130 requires a new licence key because the licence term is V1R3. Note that the VxRy values refer to aXes Version and Release and are not related to the operating system version and release values.

You will also require a new licence key if you need to run more sessions than you have purchased or if you need to run aXes on additional processors.

Each aXes server will send a message warning of impending licence expiry. This is a severity-30 message and is sent to the QSYSOPR message queue. The default is for the message to be sent 45 days prior to the expiry date. The message is sent once every day until a new licence key is installed. Customers are expected to obtain a replacement licence key within this period.

In addition, the licence manager (part of IBM i) sends messages when the licensed usage limit has been exceeded. A 30-day grace period exists in which the system will allow the usage limit to be exceeded by up to 50%. Messages will be sent during the grace period for each usage that exceeds the limit. The grace period starts from the first excess usage. The aXes servers will not stop after the grace period has expired but they will not allow excess users to obtain a session. A customer is expected to either reduce usage to within the licensed limits, or purchase additional licenses to cover the additional usage within the grace period.

LANSA will not be responsible for a customer who experiences production issues as a result of failing to obtain an appropriate licence key within the notification period.

The operating system allows licence usage to be tracked using the Work with Licence Information (WRKLICINF) command. This command shows all the available information about a licence including the current usage. The command is shipped by IBM with *PUBLIC *EXCLUDE authority so a system administrator will need to authorise appropriate users to the command.

Content © LANSA Group
All Rights Reserved

Because of the *EXCLUDE authority on WRKLICINF, aXes provides the Display aXes Licence Information (DSPAXESLIC) command. This command shows the type of licence for each feature, the usage limit, current usage, and peak usage. This is sufficient information for a customer to monitor usage. If the user is suitably authorised a function key is activated which will run the WRKLICINF command. A user will need authority to WRKLICINF if they want to see which jobs are actually using the licences.

Troubleshooting Licenses

If you experience unexpected behaviour related to licensing (including licensing messages) use DSPAXESLIC and WRKLICINF commands to determine actual license usage. Compare information displayed by the commands with your licences, and if you find differences in the information you should determine the cause. In most cases, the cause will be either:

- Attempting to start more sessions than licensed.
- Attempting to run on more processors than licensed.

It is possible, although unlikely, that licences might not be released when sessions end or when servers terminate. This would cause the usage to continually increase, eventually exceeding the limit. If this is the case it will be apparent from the DSPAXESLIC panel—the Count value will be equal to or higher than the Limit value and also higher than the number of sessions or processors the customer thinks are in use.

If this situation of licences not being released does occur, it can be corrected by using the Release aXes Licence (RLSAXESLIC) command. This command requires that the aXes instance for which licences are being released is not active. The steps would be:

- Ensure users end their aXes sessions
- Stop the aXes servers
- Release licences
- Start the aXes servers
- Allow users on again

The elapsed time for stopping and restarting the servers is short. Stopping and restarting can occur during a lunch break, at end-of-day, or even during the day. The steps are similar to normal problem resolution for device or application issues.

If the problem occurs frequently then suitable diagnostic material should be collected and forwarded to LANSA for investigation.

The DSPAXESLIC and RLSAXESLIC commands are described in the aXes Reference and on-line help.

Content © LANSA Group
All Rights Reserved

Section 2 - Managing aXes Security

This section introduces you to security considerations for using the aXes servers.

Server Security

aXes Application Server

aXes Application Server supports HTTP Basic Authentication. Basic authentication provides an optional challenge-response authentication mechanism, which can be used by an aXes Application Server instance to challenge a client request and allow a client to provide authentication information.

aXes Data Explorer Server

Table security is handled by the target data source database manager so users cannot access data in table (or views) to which they are not authorised.

aXes Data Explorer Server requires a valid user ID and password to access the target data source. For DB2/400 this means that incorrect passwords may result in a disabled user profile if the number of failed sign-on attempts exceeds the value specified in the QMAXSIGN system value.

aXes Spool File Server

Spooled file security is handled by IBM i so a user accessing spooled files via aXes Spool File Server cannot perform functions that they cannot also perform using traditional system access.

aXes Spool File Server requires a valid IBM i user profile and password to access spooled files. This means that incorrect passwords may result in a disabled user profile if the number of failed sign-on attempts exceeds the value specified in the QMAXSIGN system value.

aXes Terminal Server

Terminal security is handled by IBM i. Normal device security considerations apply such as the user attempting to sign on to the device must be authorized to use that device, and users with *ALLOBJ or *SERVICE special authority are subject to the setting of the QLMTSECOFR system value.

aXes Terminal Server requires a valid IBM i user profile and password. This means that incorrect passwords may result in a disabled user profile if the number of failed sign-on attempts exceeds the value specified in the QMAXSIGN system value.

Users can be defined to aXes Terminal Server by listing them in the Users section of the aXes Terminal Server configuration file. See the Terminal Server Configuration section in the aXes Reference for a detailed explanation of the syntax.

aXes Terminal Server will only perform administration functions, such as displaying statistics or user sessions, if it has received an authentication header from the browser. The application server manages the authentication challenge-response process.

When using the aXes Application Server, the **–authpath** switch is used to inform aXes Application Server which Access Control List should be used for authentication. See FastCGI Directives in the aXes Reference for detailed information about this switch.

Content © LANSA Group
All Rights Reserved

Basic Authentication Scheme

The basic HTTP authentication scheme is based on the model that the client must authenticate itself with a user-ID and a password for each realm. A realm allows protected resources to be partitioned into a set of protection spaces referenced by name. Each realm can have its own authentication scheme and authorization information. A realm for aXes Application Server maps to a directory path. The aXes Application Server will service the request only if it can validate the user ID and password for the protection space of the Request-URI. The aXes Application Server uses Access Control Lists (ACL) to verify user ID and password combinations.

aXes Access Control Lists

An aXes Access Control List is an object that contains the names of authorized users and encrypted passwords. It is used to control access to a directory path. You can protect specific directories, and therefore the files in those directories, by creating an ACL in the directory you want to secure. An empty ACL effectively denies access to the directory.

You can add the names of users and an associated password to the ACL to allow those users access to the directory and its files. The aXes Application Server will issue an authentication challenge to any browser requesting a URL that maps to a file in a protected directory. It will compare the user name and password provided by the user to the entries in the ACL. Access will be denied to users who do not have a valid entry in the ACL.

> Directory protection is not inherited by sub-directories. Each directory you want to protect must have its own ACL.

Access control lists are implemented as hidden files and are created in each directory you want to protect. The name of the hidden file is **.axespwd**. The file must have a code page or CCSID attribute of 37. Passwords are stored in an encrypted form. You must use the access control list CL commands to manage these objects, although you can use stream file commands to copy them if you want to duplicate an ACL from one directory to another.

See the Managing aXes Users section in this guide for information about managing users in access control lists.

Default aXes Access Control Lists

There are two Access Control Lists (ACL) supplied with axes:

One ACL is in the **<base-dir>/ts/admin** directory and protects the aXes Terminal Server session and statistics forms. The other is in the **<base-dir>/admin** directory and protects access to the aXes Application Server statistics forms.

The default user and password for both these ACLs is **admin** and **admin**. You should change the default password using the CHGAXESUSR (Change aXes User) command.

If logging is activated, then an empty ACL is created in the specified log directory. The aXes Application Server does this to stop log files being served by the web server. If you want to serve log files then you can add user entries to this ACL for those users who are allowed to view the logs.

aXes Access Control List Commands

Control Language (CL) commands are provided on the aXes Administration Menu (GO AXESADM) to assist in managing and administering access control lists. See the command syntax diagrams in the aXes Reference for information on making the aXes commands and menus available to your job.

Content © LANSA Group
All Rights Reserved

The following control list commands are available.

Create aXes Access Control List (CRTAXESACL):

The Create aXes Access Control List command allows you to create an aXes access control list object.

Delete aXes Access Control List (DLTAXESACL):

The Delete aXes Access Control List command allows you to delete an aXes access control list object.

> To use these commands, you must have *SECADM special authority.

Creating aXes Access Control Lists

You can use the CRTAXESACL (Create aXes Access Control List) command to create an Access Control List in a specific directory.

CRTAXESACL DIR('/company/sales/forecasts')

This command will protect the contents of the specified directory. Only users who were added to the ACL would be allowed to see files in that directory in their browser.

Deleting aXes Access Control Lists

You can use the DLTAXESACL (Delete aXes Access Control List) command to delete an Access Control List in a specific directory.

DLTAXESACL DIR('/company/sales/forecasts')

This command will delete the ACL and therefore also remove the protection on the specified directory.

Content © LANSA Group
All Rights Reserved

Section 3 - Managing aXes Users

This section introduces you to the commands needed to manage users. aXes considers that there are two kinds of user — terminal users and web users.

Terminal users are normal IBM i users, that is, users with a user profile (*USRPRF) object who have the capability to sign on to IBM i. Terminal users are managed using the normal IBM i user profile commands (CRTUSRPRF, CHGUSRPRF, DLTUSRPRF, and WRKUSRPRF).

A web user is created by adding the user name and associated password to an Access Control List (ACL).

See the Managing aXes Security section in this guide for information about how to create and delete ACLs.

User Commands

Control Language (CL) commands are provided on the aXes Administration Menu (GO AXESADM) to assist in managing and administering users.

The following user commands are available.

Add aXes User	ADDAXESUSR	The Add aXes User command allows you to add a user to an aXes access control list.
Change aXes User	CHGAXESUSR	The Change aXes User command allows you to change a user in an aXes access control list object.
Remove aXes User	RMVAXESUSR	The Remove aXes User command allows you to remove a user from an aXes access control list object.
Work with aXes Users	WRKAXESUSR	The Work with aXes Users command allows you to display and work with a list of users defined in an aXes access control list. You can add, change, and remove users.

See the command syntax diagrams section in the aXes Reference for information on making the aXes commands and menus available to your job.

> To use these commands, you must have *SECADM special authority.

Adding Users to an aXes Access Control List

You can add a user to an ACL using the ADDAXESUSR (Add aXes User) command. For example:

ADDAXESUSR USER(hayashi) PASSWORD(d6wi89g)

DIR('/company/sales/forecasts')

This command will allow Hayashi-san to access files in the specified directory.

Content © LANSA Group
All Rights Reserved

Changing User Passwords

You can change a user's password in an ACL using the CHGAXESUSR (Change aXes User) command. For example:

CHGAXESUSR USER(hayashi) PASSWORD(xyz876)

DIR('/company/sales/forecasts')

This command will change the password used by Hayashi–san to access files in the specified directory.

Removing Users from an aXes Access Control List

You can remove a user from an ACL using the RMVAXESUSR (Remove aXes User) command. For example:

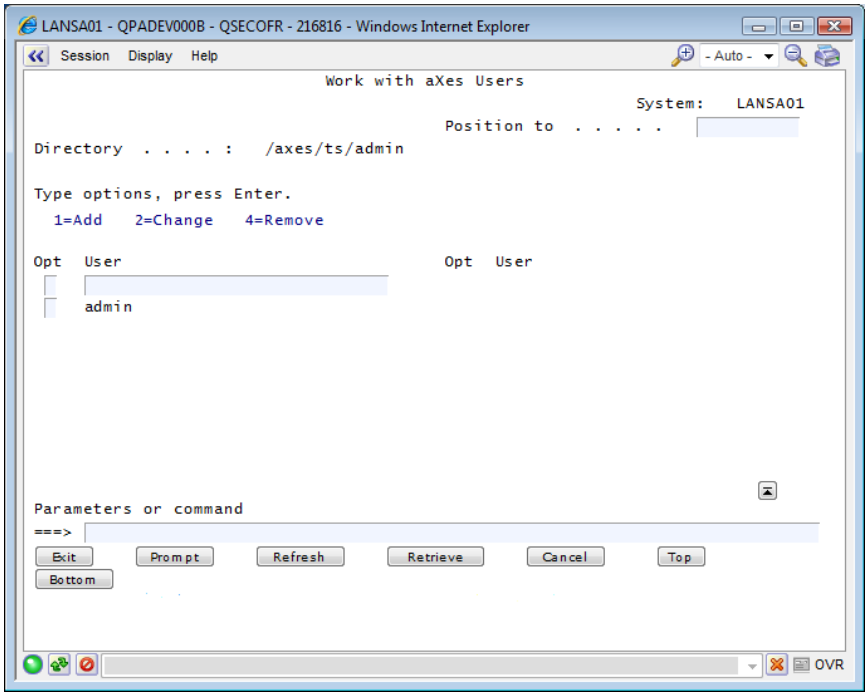
RMVAXESUSR USER(hayashi) DIR('/company/sales/forecasts')

This command will remove Hayashi-san from the ACL and therefore cause the web server to deny her access to any files in the specified directory.

Working with aXes Users

The WRKAXESUSR command displays a list of defined web server users.

This command also allows you to add, change, and remove users.



Content © LANSA Group
All Rights Reserved

Section 4 - Authorities and Permissions

This section provides information about the necessary authorities and permissions required for aXes to function correctly.

Authorities and permissions apply to:

- User Profiles
- Adopted Authority
- Exclude Authority
- aXes Application Server
- aXes Data Explorer Server
- aXes Terminal Server
- aXes Spool File Server
- aXes Gateway Interface

User Profiles

aXes provides two user profiles:

- AXES
- AXES_GROUP

The AXES_GROUP profile is used to provide Primary Group Profile (PGP) authority for stream files and directories. The AXES profile is a member of AXES_GROUP and also owns the product stream files and directories. This arrangement is used because PGP authority is automatically propagated to new objects created in the directory tree therefore AXES will inherit appropriate authority to new objects.

aXes server jobs run using the AXES user profile. Therefore, the AXES user profile must have authority to the product objects and to any directories and stream files used by the servers.

The AXES_GROUP user profile does not require any special authorities.

The AXES user profile requires ***JOBCTL** special authority to retrieve job information. During installation a random password is generated for the AXES user profile. If you want to change this password do so using the CHGUSRPRF (Change User Profile) command.

A password is required for the AXES user profile because it is used by the aXes Application Server to connect to the *LOCAL database when running Lua scripts that contain SQL statements. You may change this password to another value of your choice.

> None of the aXes servers need to know the AXES profile password but one must exist to allow it to access the *LOCAL database.
>
> If you set the password to *NONE then Lua scripts running under the aXes Application Server fail unless a user profile and password are supplied within the Lua script.
>
> The aXes Data Explorer Server will also fail when building the list of available data sources.

Content © LANSA Group
All Rights Reserved

Adopted Authority and System State

Programs that adopt authority

aXes generally relies on *PUBLIC authority to access objects, however there are some program objects that adopt authority. The following table shows programs that adopt authority.

Table 1: Programs that Adopt Authority

Object	Type	Reason
AXADOPT	*SRVPGM	Verify the password for aXes Terminal Server users
AXCHKATNKY	*PGM	Check if the job has an Attention program (ATNPGM) defined.
AXCHKPGSTK	*PGM	Check the program stack of a job for a particular program. Used to support the System Request (SysRq) function.
AXDSPMSGW	*PGM	Display the message for jobs in MSGW state.
AXENDTS	*PGM	***JOBCTL** authority required to end aXes Terminal Server
AXENDW3	*PGM	***JOBCTL** authority required to end aXes Application server
AXSTRTS	*PGM	***JOBCTL** authority required to start AXES subsystem and ***USE** authority required to AXES user profile to submit jobs under that profile
AXSTRW3	*PGM	***JOBCTL** authority required to start AXES subsystem and ***USE** authority required to AXES user profile to submit jobs under that profile
AX5250INZ	*PGM	***SAVSYS** authority required.

Programs that run in system state

There are also some programs that must run in system state to perform their functions. The following table shows the system state programs.

Table 2: Programs that Run in System State

Object	Type	Reason
AXCHKATNKY	*PGM	Check if the job has an Attention program (ATNPGM) defined.
AXCHKPGSTK	*PGM	Check the program stack of a job for a particular program. Used to support the System Request (SysRq) function.
AXDSPMSGW	*PGM	Display the message for jobs in MSGW state
AXGETCCSID	*PGM	Determine the CCSID of a display file
AXRTVVTDEV	*PGM	Determine the associated virtual device
STDERRC2	*PGM	Dump a CL program
AXSPACE	*PGM	Insert space into a context
AXINDEX	*PGM	Insert index into a context

Content © LANSA Group
All Rights Reserved

Object	Type	Reason
AXRUNCMD	*PGM	Run remote command
AXCMDCLNUP	*PGM	Clean up remote command
AX5250INZ	*PGM	InterSession support
AX5250EXIT	*PGM	InterSession support
AX5250ORSV	*PGM	InterSession support
AX5250OSJE	*PGM	InterSession support
AX5250OSPM	*PGM	InterSession support
AX5250RQT	*PGM	InterSession support
AX5250SQ	*PGM	InterSession support
AX5250DEVD	*PGM	InterSession support
AXINLPGM	*PGM	InterSession support
AXGRPPGM	*PGM	InterSession support

aXes programs you can alter or delete

If having programs which adopt authority or have system state set concerns you then you may alter or delete some of the programs as long as you understand the effect that will have on the product. The following table shows programs you may delete.

Table 3: Programs You Can Delete and the Consequences

Programs	Consequences of Deletion
AXCHKATNKY	Deleting the program will mean that aXes can no longer protect a user from pressing the Attention (Attn) key when it is not appropriate. Because this can cause the terminal session to appear to hang aXes will no longer allow the Attention key to be pressed.
AXCHKPGSTK	Deleting the program will mean that aXes can no longer protect a user from pressing the Attention (Attn) key or System Request (SysRq) key when it is not appropriate. Because this can cause the terminal session to appear to hang aXes will no longer allow the Attention key or System Request key to be pressed.
AXDSPMSGW	Deleting the program will mean that you can no longer answer messages for aXes servers which are in a MSGW state from the Work with aXes Servers display.
STDERRC2	Deleting the program will mean that you can no longer get a diagnostic dump from CL programs that may end in error.

Content © LANSA Group
All Rights Reserved

Remove adopt authority for these aXes programs

You may change the following programs to remove authority adoption:

AXENDTS	You must ensure that users of the End aXes Terminal Server (ENDAXESTS) command have ***JOBCTL** special authority.
AXENDW3	You must ensure that users of the End aXes Application Server (ENDAXESW3) command have ***JOBCTL** special authority.
AXSTRTS	You must ensure that users of the Start aXes Terminal Server (STRAXESTS) command have ***JOBCTL** special authority and that they have ***USE** authority to the **AXES** user profile.
AXSTRW3	You must ensure that users of the Start aXes Application Server (STRAXESW3) command have ***JOBCTL** special authority and that they have ***USE** authority to the **AXES** user profile.

Do not remove adopt authority for these aXes programs

You must not alter or delete the following programs because authority adoption is fundamental to the correct working of the product.

AXADOPT

aXes programs not to delete

If you use the aXes Intersession feature do not delete any program in the following list.

AXRUNCMD AXCMDCLNUP AX5250INZ AX5250EXIT AX5250ORSV AX5250OSJE AX5250OSPM AX5250RQT AX5250SQ AX5250DEVD AXINLPGM AXGRPPGM AXSPACE AXINDEX	Deleting any of these programs will mean you can no longer use the aXes Intersession capability.

Exclude authority

The following objects are shipped with *PUBLIC *EXCLUDE authority:

Object	Type	Description
ENDAXESTS	*CMD	End aXes Terminal Server
ENDAXESW3	*CMD	End aXes Application Server
STRAXESTS	*CMD	Start aXes Terminal Server

Content © LANSA Group
All Rights Reserved

Object	Type	Description
STRAXESW3	*CMD	Start aXes Application Server
AXRUNCMD	*PGM	aXes Intersession support
AX5250INZ	*PGM	aXes Intersession support
AX5250EXIT	*PGM	aXes Intersession support
AX5250ORSV	*PGM	aXes Intersession support
AX5250OSJE	*PGM	aXes Intersession support
AX5250OSPM	*PGM	aXes Intersession support
AX5250RQT	*PGM	aXes Intersession support
AX5250SQ	*PGM	aXes Intersession support

UNIX-style Permissions

Most of the files served by the aXes Application Server, including files used to describe panels for aXes Terminal Server and script files containing Lua source, are stored as stream files in the Integrated File System (IFS). They have a different level of data authorization (called permissions) from normal IBM i data authorities.

This section is intended to give you an understanding of how these permissions are managed.

You can find information specific to permissions on the IBM i in the Information Centre.

For a generic UNIX view of permissions you can find information on the Internet, with thousands of articles and tutorials available. Use a search engine and search for **UNIX permissions**.

Overview

Permissions are associated with each file and directory. For files, permissions control which users have access to read, write, and execute them. For directories, permissions control which users can view directory contents, create or delete files in a directory, or navigate into that directory.

Any of the permissions can be applied in any combination to any users, and there are three classifications for users: **Owner**, **Group**, and **Other**.

Users and Access

The concept of users is an abstract classification; referring not to a specific user, but to access:

Owner (also Admin or you)	The owner of the file has access.
Group	Other members of the same classification as the owner have access.
Other (also public or everyone)	Everyone else has access. This is equivalent to *PUBLIC.

Content © LANSA Group
All Rights Reserved

Files and Permissions

Permissions for files mean exactly what they sound like:

Read	Controls whether a user can read the file.
Write	Controls whether a user can write to the file.
Execute	Controls whether a user can run the file. Used for files that are scripts or programs.

Directories and Permissions

Permissions for directories are subtly different from those for files:

Read	Controls whether a user can view the contents of the directory, that is, display a list of the files in the directory.
Write	Controls whether a user can create new files or delete existing files in the directory. Users with write access to a directory can delete files in that directory, even when the users do not have write permission to the files.
Execute	Controls whether the user can navigate into that directory. Navigate means changing from one directory to another. It also controls whether the user can navigate through a directory and its sub-directories. That is, in order to manipulate the contents of a sub-directory, the user must have execute permission to all directories in the path.

How Permissions are Calculated

Permissions are stored in 9 bits split into 3 groups of 3 bits each. Each bit represents a specific permission for a specific user classification. The full range of permissions can be represented as **rwxrwxrwx** where the first group holds permissions for the owner, the second group holds permissions for the group, and the final group holds public permissions. This arrangement lends itself to a convenient, if arcane, method of describing permissions using octal (or base 8) notation. Each permission is assigned a numeric value: **read = 4**, **write = 2**, **execute = 1**, thus each user (owner, group, other) permission can be calculated by summing the permissions to be granted.

For example, to grant permission to **read** and **write**, the permission is **4+2=6** and to grant permission to **read**, **write**, and **execute** is **4+2+1=7**, and so on.

How Permissions are Granted

UNIX-style permissions are commonly set with the chmod shell command. On the IBM i this command can be run via QSHELL and it has the general format:

```
chmod mode filename
```

Where **chmod** is the command, **mode** is the numerical expression of the permissions, and **filename** is the name of the file or directory.

For example:

```
chmod 644 /sales/forecasts/2002data.csv
```

This command will grant read and write permission to the owner, read permission to the group, and read permission to *PUBLIC.

Content © LANSA Group
All Rights Reserved

The chmod command also supports a mnemonic form that would allow the previous example to be written as:

chmod u=rw,g=r,o=r /sales/forecasts/2002data.csv

Since we are discussing IBM i rather than a UNIX system, there is also a command interface to changing permissions. The command is the Change Authority (CHGAUT) command and it provides a more natural mechanism. For example:

Files	CHGAUT OBJ('/w3/dir/*') USER(*PUBLIC) DTAAUT(*R)
Directories	CHGAUT OBJ('/w3/dir') USER(*PUBLIC) DTAAUT(*RX) Or CHGAUT OBJ('/w3/dir') USER(*PUBLIC) DTAAUT(*R)
aXes Gateway Interface (AGI)	CHGAUT OBJ('/w3/agi/sample.agi') USER(*PUBLIC) DTAAUT(*X)

FTP client utilities generally provide a means of setting the permissions, either by the numeric representation of the permissions, or with a check-box selection.

In summary object modes should be as follows.

Objects	Mode
Data files	644 (rw-r--r--)
Directories	755 (rwxr-xr-x) to allow indexing 711 (rwx--x--x) to prevent indexing
aXes Gateway Interface (AGI) programs	755 (rwxr-xr-x) or 711 (rwx--x--x)

aXes Application Server

The aXes Application Server requires that ***PUBLIC** has **read (*R)** authority to the files being served. aXes Application Server explicitly checks for public read permission. This requirement applies to all data files served by aXes Application Server such as HTML files or images, and is imposed to ensure that no file is ever delivered to the web unless it is accessible to *PUBLIC.

aXes Application Server also requires that **execute (*X)** permissions are not set for data files. A file marked as executable but does not match the AGI pattern might be a script or program that was accidentally left in the wrong directory. Allowing a browser to fetch the contents of an executable file is considered a security exposure and thus is not allowed.

aXes Data Explorer Server

aXes Data Explorer has no special authority or permission requirements.

aXes Spool File Server

The aXes Spool File Server requires that ***PUBLIC** has **read (*R)** authority to the files being served. aXes Spool File Server explicitly checks for public read permission. This requirement applies to all data files served by aXes Spool File Server such as HTML files or images, and is imposed to ensure that no file is ever delivered to the web unless it is accessible to *PUBLIC.

aXes Spool File Server also requires that ***PUBLIC execute (*X)** permissions are set for files that contain Lua script, however ***PUBLIC** must not have ***X** permission for files that do not contain Lua script.

Content © LANSA Group
All Rights Reserved

aXes Terminal Server

aXes Terminal Server has no special authority or permission requirements.

aXes Gateway Interface

aXes Gateway Interface (AGI) scripts run with the authorities of the user profile of the aXes Application Server job. In addition, AGI scripts must have ***PUBLIC *X (execute)** authority but they must not have ***PUBLIC *R (read)** authority.

AGI scripts must exist in a designated directory. The reason for these specific settings is so that scripts cannot be retrieved or served by the aXes Application Server and can only be run by the server. This is done in order to protect the contents of script files from being viewed in a web browser.

Either the **AXES** or **AXES_GROUP** profile must have **read (*R)** permission to the file.

Common Gateway Interface

Common Gateway Interface (CGI) scripts are subject to the same permission requirements as AGI scripts. In addition, either the **AXES** or **AXES_GROUP** profile must have **read** and **execute (*RX)** permission to the file so it can be spawned successfully.

FastCGI Servers

FastCGI servers provide a way for you to extend the function of a web server by writing your own programs to handle requests from web browser clients. FastCGI servers are supported directly by the aXes Application Server and also by the IBM HTTP server if you configure the FastCGI service adapter supplied with the aXes product.

FastCGI servers run as separate jobs and are classed as internal servers or as external servers. Internal servers are started from within a FastCGI enabled web server. External servers are started by some other means.

Internal servers are started as batch-immediate jobs and inherit their authorities from the process that started them.

External servers are submitted as batch jobs and use the authorities of the user profile under which they are started.

Both forms of FastCGI server may dynamically change the current user profile by using the appropriate IBM APIs if they are authorized to the APIs and to the new user profile.

Once a FastCGI server is started it is just another IBM i job and is subject to the same object level security as any other job on the system.

Content © LANSA Group
All Rights Reserved

Section 5 - Understanding FastCGI

This section introduces you to FastCGI and its use by the aXes servers. Complete information about FastCGI can be found at www.fastcgi.com but this section provides a sufficient overview to intelligently configure the aXes servers.

Read this section before configuring a FastCGI External Server with aXes.
You cannot use aXes eXtensions when you configure aXes with a FastCGI External Server.

About the Common Gateway Interface (CGI)

CGI is the Common Gateway Interface, which is a standard protocol that defines how a web server interfaces with external applications. This allows the web server to pass requests from a client web browser to the external application. The web server receives output from the application and passes the output to the browser.

FastCGI is a language-independent, scalable, open, extension to CGI that provides high performance without the limitations of server-specific APIs.

aXes FastCGI Servers

The aXes servers are FastCGI servers. This means they are written to the FastCGI interface specification. Therefore understanding FastCGI will be helpful in managing the aXes servers.

FastCGI servers come in two types; Internal FastCGI Servers and External FastCGI Servers.

Both server types behave the same way. The difference is in how they are defined, started, and how they communicate with the web server.

Both server types are defined in the web server configuration file. Internal servers are defined using the FastCGIServer directive. External servers are defined using the FastCGIExternalServer directive.

Internal servers are started automatically by the web server and use a UNIX-domain socket (essentially a special kind of stream file) to communicate. Internal servers must run on the same system as the web server.

External servers are started by a user-command. They require a TCP/IP network socket for communication. External servers also need to know the IP address and port on which they are to listen for web server requests. Although external servers may run on the same system as the web server, because they use network sockets they can also run on a system different from the one on which the web server is running. External servers also have the advantage that they can be started and stopped independently of the web server. See the configuration scenarios in the aXes Reference for sample configurations.

aXes ships with Internal FastCGI definitions in the web server configuration files. This means that when the web server is started it automatically starts the other aXes servers. It also ships with example External FastCGI Server definitions and the Work with aXes Servers (WRKAXESSVR) command shows predefined External FastCGI Server definitions for all the aXes servers. The aXes Application Server is defined with AUTOSTART(*YES) and the external servers are defined with AUTOSTART(*NO).

Operating aXes Application Server

In normal operation you would start only the aXes Application Server and let it automatically start the other servers as Internal FastCGI Servers. These internal servers are started

Content © LANSA Group
All Rights Reserved

regardless of the Auto-start setting shown in the WRKAXESSVR panels. External FastCGI Servers are intended for advanced users and should only be used when the web server must be on a different system, or when the aXes servers need to be started and stopped independently.

The aXes Application Server can be started in one of three ways:

1. The Start aXes Application Server (STRAXESW3) commands.
2. Option 9=Start on the WRKAXESSVR panel.
3. Automatically when the AXES subsystem is started if the server instance is defined with AUTOSTART(*YES) on the WRKAXESSVR panel.

The STRAXESW3 command will start the application server. It will also start other aXes servers if they have been defined as internal servers in the application server configuration file.

External aXes server definitions can also be started in one of three ways:

1. The various start aXes server (STRAXESXX) commands.
2. Option 9=Start on the WRKAXESSVR panel.
3. Automatically when the AXES subsystem is started if the server instance is defined with AUTOSTART(*YES) on the WRKAXESSVR panel.

The STRAXES**XX** (where **XX** is DE, TS, or WS) commands will start an External FastCGI Server instance. Each instance requires a corresponding FastCGIExternalServer directive in the configuration file of the application server assigned to handle requests and responses for the instance. The IP address and port supplied on the start command must match the information specified in the directive.

The various aXes servers are optional components of the product. If they are not installed an error will occur when attempting to start them. An error message will be found in the job log of the application server job for each internal server that failed to start. The error can be corrected by either installing the missing component (and obtaining a suitable licence), or by commenting-out the appropriate FastCGIServer directive in the application server configuration file.

Configuring FastCGI External Server

The aXes FastCGI External Server feature may be used to achieve this type of configuration:

Figure 1: aXes FastCGI External Server

In this configuration the aXes Terminal Server runs on an IBM i server located behind a firewall. This provides an extra layer of security by not directly exposing the IBM i server to the Internet. However, this configuration prevents the use of aXes eXtensions.

Reverse proxy is a superior option for implementing this type of configuration.

A reverse proxy solution:

- Provides the same level of security with the IBM i server behind a firewall.
- Simplifies configuration and maintenance.
- Allows the use of aXes eXtensions.

Content © LANSA Group
All Rights Reserved

> You cannot use aXes eXtensions when you configure aXes with a FastCGI External Server.

Another advantage of a reverse proxy configuration is the separation of network configuration from aXes configuration. No FastCGI configuration is required on the proxy server outside the firewall. Network administrators can manage and maintain the reverse proxy configuration without having to learn how to configure aXes.

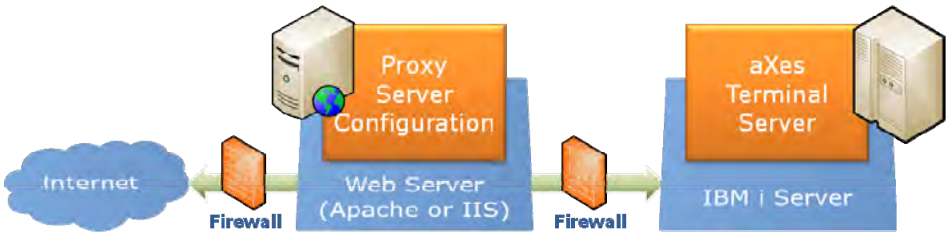


Figure 2: aXes with a Proxy Server

Please talk to your aXes reseller about using a reverse proxy solution before attempting to implement a FastCGI External Server solution.

Content © LANSA Group
All Rights Reserved

Section 6 - Managing aXes Servers

This section introduces you to the commands that allow you to control and manage aXes servers. The aXes servers are:

- aXes Application Server (aXes-W3)
- aXes Terminal Server (aXes-TS and aXes-TS2)
- aXes Data Explorer Server (aXes-DE)
- aXes Spool File Server (aXes-WS)

The aXes servers require a web server to handle connection from a browser and can use the aXes Application Server or Apache Web Server (using the FastCGI module).

LANSA recommends that you use the aXes Application Server.

See the command syntax diagrams in the aXes Reference for information on making the aXes commands and menus available to your job.

Defining aXes Server Instances

All aXes servers allow you to define server instances where you can associate a name with a specific configuration file and override various server attributes.

> All releases of aXes will have default servers defined with the same name (AXES).
>
> You may use the ADDAXESSVR command to define instances for the new release that have different instance names from those used by the existing release.

Server instances are created using the following commands:

ADDAXESW3	Add aXes Application Server
ADDAXESDE	Add aXes Data Explorer Server
ADDAXESWS	Add aXes Spool File Server
ADDAXESTS	Add aXes Terminal Server

aXes servers can be configured in the following ways:

- As an internal FastCGI server started by aXes Application Server
- As an external FastCGI server

Internal servers are started automatically by the aXes Application Server and use a private communication channel between themselves and the associated application server. Internal servers also run on the same system as their associated application server. This is the easiest way to configure the aXes servers because they will start and stop automatically when aXes Application Server is started and stopped.

External servers use sockets to communicate between themselves and their associated web server such as aXes Application Server. External servers can also run on a system that is remote from their associated web server. External servers provide a way to share one or more FastCGI servers with one or more web servers. External servers provide a more flexible way of setting up the aXes servers because they can be started and stopped independently, however the configuration is more complicated.

> See the Application Server Configuration section in the aXes Reference for detailed explanations of the directive and command syntax used to define aXes server instances for Terminal Server, Data Explorer Server, Spool File Server and aXes Application Server.

Content © LANSA Group
All Rights Reserved

aXes Terminal Server

Internal server configuration

The FastCGI section of the aXes Application Server configuration file is used to define aXes Terminal Server as an internal server. An example directive is:

```
FastCGIServer "/axests" \
–program "AXESTS.PGM '/axes/configs/aXesTS.conf'" –mt \
-idle-timeout 45 –authpath='admin=/axests/admin/**'
```

This directive defines an internal FastCGI aXes Application Server which will be started when the aXes Application Server instance is started. Any URLs starting with **/axests** will be passed to this server, responses must occur within 45 seconds, and the administration functions in the URL **/axests/admin** are protected by the access control list found in the **admin** directory under the server root directory.

External server configuration

The STRAXESTS (Start aXes Terminal Server) command is used to start aXes Terminal Server as an external server. For example:

```
STRAXESTS INSTANCE(AXESEXTERN)
CFG('/axes/configs/axests.conf')
INTNETADR(*ANY 6666)
```

This directive will start an instance of the aXes Terminal Server using a job name of **AXESEXTERN** which will listen for requests on port **6666**. One or more web servers can reference this server in their configuration file. The web servers may be running on remote systems.

To allow a local aXes Application Server to forward requests to this external server you could define the following directive in the configuration file:

```
FastCGIExternalServer "/axests" –appConnTimeout 30 \
    -idle-timeout 30 –host ":6666"
```

If you want to allow remote web servers to use this external terminal server you must specify the IP address or name of the system on which the terminal server is running.

To allow a remote Apache server to forward requests to this terminal server you could define the following directive in the Apache configuration file:

```
FastCGIExternalServer "axests" –host remotehost:6666
```

Where **remotehost** is the IP address or name of the system on which the Terminal Server is running.

You must also configure the FastCGI module for this instance of the Apache server using directives similar to the following:

```
LoadModule fastcgi_module modules/mod_fastcgi.dll
AddModule mod_fastcgi.c
```

The actual syntax will vary according to the version of Apache and the platform on which Apache is running. See the configuration scenarios appendix in the aXes Reference and your Apache configuration documentation for more information.

Content © LANSA Group
All Rights Reserved

aXes Data Explorer Server

Internal server configuration

The FastCGI section of the aXes Application Server configuration file is used to define aXes Data Explorer Server as an internal server. For example:

```
FastCGIServer "/axesde" \
–program "AXESDE.PGM '-config=/axes/configs/aXesDE.conf'" \
-idle-timeout 45
```

This directive will define an internal FastCGI aXes Data Explorer Server which will be started when the aXes Application Server instance is started. Any URLs starting with /axesde will be passed to this server and responses from the server must occur within 45 seconds.

External server configuration

The STRAXESDE (Start aXes Data Explorer Server) command is used to start aXes Data Explorer Server as an external server. For example:

```
STRAXESDE INSTANCE(AXESEXTERN)
  CFG('/axes/configs/axesde.conf')
  INTNETADR(*ANY 6666)
```

This directive will start an instance of the aXes Data Explorer using a job name of **AXESEXTERN** which will listen for requests on port **6666**. One or more web servers can reference this server in their configuration file. The web servers may be running on remote systems.

To allow a local aXes Application Server to forward requests to this external server you could define the following directive in the configuration file:

```
FastCGIExternalServer "/axesde" –appConnTimeout 30 \
    -idle-timeout 30 –host ":6666"
```

If you want to allow remote web servers to use this external server you must also specify the IP address or name of the system on which the terminal server is running.

```
FastCGIExternalServer "axesde" –host remotehost:6666
```

Where **remotehost** is the IP address or name of the system on which the Data Explorer is running.

You must also configure FastCGI support for the remote web server. See the documentation supplied with your remote web server or the FastCGI web site.

aXes Spool File Server

Internal server configuration

The FastCGI section of the aXes Application Server configuration file is used to define aXes Spool File Server as an internal server. For example:

```
FastCGIServer "/axesws" \
–program "AXESWS.PGM '-config=/axes/configs/aXesWS.conf'" \
-idle-timeout 45
```

This directive will define an internal FastCGI aXes Data Explorer Server which will be started when the aXes Application Server instance is started. Any URLs starting with **/axesws** will be passed to this server and responses from the server must occur within **45 seconds**.

Content © LANSA Group
All Rights Reserved

External server configuration

The STRAXESWS (Start aXes Spool File Server) command is used to start aXes Spool File Server as an external server. For example:

```
STRAXESWS INSTANCE(AXESEXTERN)
  CFG('/axes/configs/axesws.conf')
  INTNETADR(*ANY 6666)
```

This command will start an instance of the aXes Spool File Server using a job name of AXESEXTERN which will listen for requests on port 6666. One or more web servers can reference this server in their configuration file. The web servers may be running on remote systems.

To allow a local aXes Application Server to forward requests to this external server you could define the following directive in the configuration file:

```
FastCGIExternalServer "/axesws" –appConnTimeout 30 \
    -idle-timeout 30 –host ":6666"
```

If you want to allow remote web servers to use this external server you must also specify the IP address or name of the system on which the terminal server is running.

```
FastCGIExternalServer "axesws" –host remotehost:6666
```

Where **remotehost** is the IP address or name of the system on which the aXes Spool File Server is running.

You must also configure FastCGI support for the remote web server. See the documentation supplied with your remote web server or the FastCGI web site.

Displaying aXes Server Instances

Defined server instances and active ad-hoc instances can be displayed using the WRKAXESSVR (Work with aXes Server) command.

Option 5=Display will show you details about the instance definition.

Starting and Stopping aXes Servers

> All servers that listen on TCP/IP ports for incoming requests must listen on unique ports.
>
> You can define the port numbers in an instance definition or on the server start command.

Starting the Servers

Servers can be started using the appropriate start command.

aXes Application Servers

aXes Application Servers are started using the STRAXESW3 (Start aXes Application Server) command. aXes Application Servers can be configured to automatically start the other aXes servers as internal FastCGI servers.

aXes Application Servers and external FastCGI Servers can also be started using option 9 of the WRKAXESSVR (Work with aXes Server) command.

aXes Terminal Server

aXes Terminal Server external FastCGI servers are started explicitly using the STRAXESTS (Start aXes Terminal Server) command or implicitly by the STRAXESW3 or STRTCPSVR (Start TCP/IP Server) command.

Content © LANSA Group
All Rights Reserved

aXes Data Explorer Server

aXes Data Explorer Server external FastCGI servers are started explicitly using the STRAXESDE (Start aXes Data Explorer Server) command or implicitly by the STRAXESW3 command.

aXes Spool File Server

aXes Spool File Server external FastCGI servers are started explicitly using the STRAXESWS (Start aXes Spool File Server) command or implicitly by the STRAXESW3 command.

> An implicit start happens when an instance of an aXes Application Server has an aXes server defined as an internal FastCGI server.
>
> You cannot use the STRAXESxx commands to start or restart an internal FastCGI server.

Stopping the Servers

Servers can be stopped using the appropriate end command.

aXes Application Servers

aXes Application Servers are stopped using the ENDAXESW3 (End aXes Application Server) command. Servers that have been automatically started by the aXes Application Server are automatically stopped when the aXes Application Server ends.

aXes Terminal Server

aXes Terminal Server external FastCGI servers are stopped explicitly using the ENDAXESTS (End aXes Terminal Server) command or implicitly by the ENDAXESW3 or ENDTCPSVR (End TCP/IP Server) command.

aXes Data Explorer Server

aXes Data Explorer Server external FastCGI servers are stopped explicitly using the ENDAXESDE (End aXes Data Explorer Server) command or implicitly by the ENDAXESW3.

aXes Spool File Server

aXes Spool File Server external FastCGI servers are stopped explicitly using the ENDAXESWS (End aXes Spool File Server) command or implicitly by the ENDAXESW3.

All Servers

Servers can also be ended using option 10 of the WRKAXESSVR (Work with aXes Server) command.

> You must use the end command from the same release as the start command used to start the server.

When to Stop and Restart aXes Servers

> You should stop and restart your aXes Server instances at least once per week and preferably once per day.
>
> The stop and restart process typically takes around 30 seconds to complete.

Many IBM i systems use an overnight window in which interactive users are prevented from using the system while backups and other overnight batch processing are performed. Typically the QINTER subsystem is terminated during this process to prevent interactive user access. The aXes server instances should be stopped and restarted as part of this nightly batch process.

If no nightly batch processing window is available you should schedule a stop and restart of all your aXes server instances at least once per week.

Content © LANSA Group
All Rights Reserved

If you restart your interactive subsystem(s) or your TCP/IP communication facilities you should similarly restart your aXes server instances.

Starting aXes Servers when the System Starts

aXes server instances can be started when the system starts by setting the server instance auto-start attribute to *YES using the appropriate add or change command.

Use these commands to set the auto-start attribute for a new instance:

ADDAXESW3 - Add aXes Application Server,

ADDAXESTS - Add aXes Terminal Server,

ADDAXESDE - Add aXes Data Explorer Server, and

ADDAXESWS - Add aXes Spool File Server.

Use these commands to set the auto-start attribute for existing instance:

CHGAXESW3 - Change aXes Application Server,

CHGAXESTS - Change aXes Terminal Server,

CHGAXESDE - Change aXes Data Explorer Server, and

CHGAXESWS - Change aXes Spool File Server.

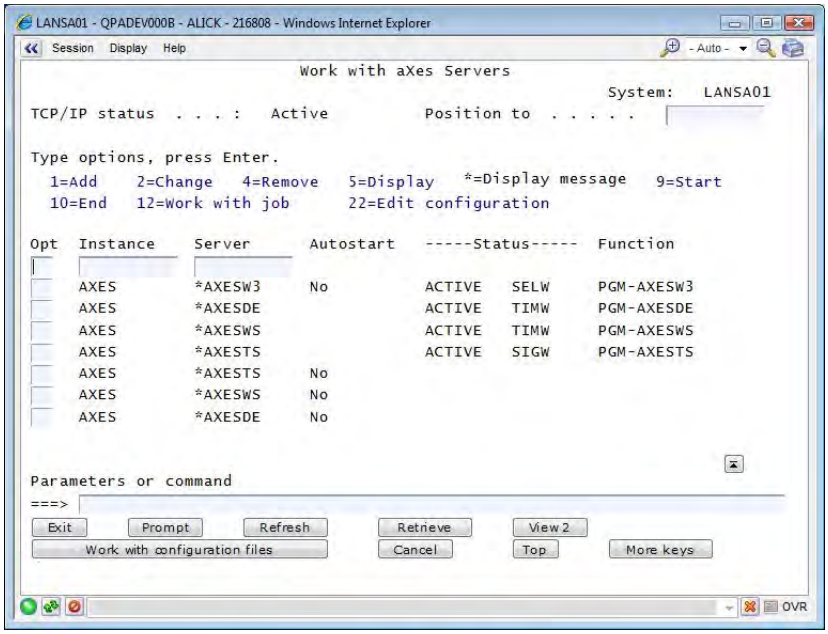
You must also start the AXES subsystem from the system start-up program. The system start-up program is named in the QSTRUPPGM system value. Edit the source for the start-up program and add a STRSBS (Start Subsystem) command to start the AXES subsystem from the installed product library.

Ensure you start the AXES subsystem after TCP/IP is successfully started.

> You cannot automatically start servers from multiple releases. You must choose one release and start its supplied subsystem. To start servers from a different release, you should use the start command from that release.

Working with aXes Server Instances

The WRKAXESSVR command displays a list of defined server instances and active ad-hoc instances. This command also allows you to add, change, display, and remove server instances, start and end servers, display and answer server messages, and work with the server job.



Content © LANSA Group
All Rights Reserved

Section 7 - Backup and Recovery

This section provides guidelines for saving aXes components and recovering in the event of a disaster. You use IBM i commands to perform the backup and recovery tasks.

About Backing up aXes

aXes server instances may contain data and programs that are vital to the ongoing operation of your business. For example, web interfaces may rely on AGI or CGI scripts that exist only within the aXes directory structure.

To protect data and programs from disasters (such as a site loss or hardware failure) and from human error (such as accidentally deleting an important script) you should develop a plan for regularly backing up the information in aXes.

Your backup plan should include:

- Objects that change infrequently, such as aXes programs
- Objects that change often, such as web pages and script files

aXes takes advantage of the IBM i single-level store architecture. aXes programs and stream files are spread across all disk units, along with other IBM i objects. The IBM i operating system automatically manages the allocation of disk space for you. To back up information in IBM i you back up logically (by object name), not physically (by disk unit). To plan a backup strategy, you must understand the logical location of your aXes objects.

Backup Tips

For information about planning a backup strategy for your system see the IBM i Backup and Recovery (SC41-5304) manual supplied with your IBM i software. You may also want to refer to the descriptions of using the Save menu and using the SAVxxx and RSTxxx commands.

Libraries and Directories for aXes

The aXes program code is held in IBM i libraries. The terminal components are held as stream files. Web components such as HTML pages and AGI scripts are also stored as stream files.

IBM i Library Structure

A default installation of aXes creates the following libraries:

AXES	The aXes base product library. The actual name depends on the values specified when aXes was installed.
AXES29nn	A library containing objects that are specific to your national language.

IFS Root File System

The integrated file system (IFS) is a part of IBM i that supports stream files and storage management similar to personal computer and UNIX operating systems, while providing an integrating structure over all information stored in IBM i.

Content © LANSA Group
All Rights Reserved

When you install aXes, a number of directories are created in the IFS root file system. They are:

axes	This directory is the root directory for the product. The actual name depends on the values specified when aXes was installed.
axes/admin	This directory controls access to the aXes Application Server administration and configuration forms.
axes/agi	This directory contains the Lua script files.
axes/configs	This directory contains the default configuration files.
axes/de	This directory contains the aXes Data Explorer Server files.
axes/de/images	This directory contains the aXes Data Explorer Server images.
axes/de/lua	This directory contains the aXes Data Explorer Server script files.
axes/de/skins	This directory contains the aXes Data Explorer Server skins.
axes/docs	This directory contains the product documentation.
axes/errors	This directory contains customized HTTP error pages.
axes/images	This directory contains image files (i.e., JPEGs and GIFs, etc.).
axes/logs	This directory contains HTTP log information and trace output.
axes/samples	This directory contains examples.
axes/ts	This directory contains aXes Terminal Server files.
axes/ts/admin	This directory contains the aXes Terminal Server administration and configuration forms.
axes/ts/skins	This directory contains the aXes Terminal Server skins and images.
axes/wba	This directory contains the aXes Server administration and configuration forms and images.
axes/ws	This directory contains aXes Spool File Server files.
axes/ws/images	This directory contains the aXes Spool File Server images.
axes/ws/lua	This directory contains the aXes Spool File Server script files.
axes/ws/skins	This directory contains the aXes Spool File Server skins.

Backing up aXes Programs and Product Files

After you install aXes you should save a copy of the product off-line, especially if you applied PTFs during installation. You should also save a copy of aXes periodically such as when you upgrade to a new release or apply PTFs.

You should incorporate these saves in your current backup procedures.

Backing up Changed Objects from aXes

Modified stream files such as company or user specific terminal skins, AGI or CGI scripts, or HTML pages should be saved when they change. Ideally, modified skins should be created in a separate directory structure and referenced from aXes instances. Separating your modified files from the aXes product files will make it easier to manage your changes.

Content © LANSA Group
All Rights Reserved

When you use the CRTAXESACL (Create aXes Access Control List) command you are creating a special file named .axespwd in the specified directory. This file contains user identifiers and encrypted passwords. When you use the ADDAXESUSR (Add aXes User), CHGAXESUSR (Change aXes User), or RMVAXESUSR (Remove aXes User) commands you are altering the contents of these files. You should save the Access Control List files regularly so you do not run the risk of losing access for web users.

When you use the ADDAXESTS, ADDAXESW3, CHGAXESTS, CHGAXESW3, RMVAXESTS, or RMVAXESW3 commands you are modifying records in the AXINSTANCE database file located in the aXes product library. You should save this file regularly so you do not lose server instance definitions.

Most of the values used to control the behaviour of the aXes servers are held in configuration files. By default these files are kept in the configs directory but you can store configuration files in any directory. You should save the configuration files regularly to ensure you do not lose server definition information.

The HTML skins used by the aXes terminal can be customised for specific corporate requirements. If you modify the skins then you should save them to ensure your changes are not lost.

Disaster/Recovery for aXes Components

In the event of a disaster that requires you to reload your system you can restore the product from the copies saved to the backup media.

aXes Libraries

Use the RSTLIB (Restore Library) command to restore the aXes product library, language libraries, and samples library.

Use the RSTOBJ (Restore Object) command to restore individual objects to the aXes libraries.

aXes Directories

Use the RST (Restore) command to restore directories and stream files to the aXes product directories.

Changed Objects

If you are saving changed objects in the aXes product libraries using the SAVCHGOBJ (Save Changed Objects) command you can use the RSTOBJ (Restore Object) command to restore changed objects from your incremental backup media.

If you are saving changed stream files using the SAV (Save) command you can use the RST (Restore) command to recover files and directories.

Saving changed objects only saves objects that have changed since the last full backup or changed since a specific date and time. If the object you want to restore has not changed it will not be saved to incremental media. In this case you will need to restore from the most recent full backup media.

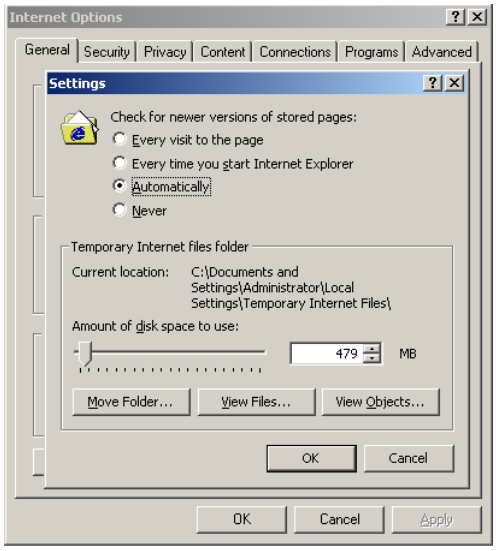
Content © LANSA Group
All Rights Reserved

Appendix A – Browser Settings

This section describes settings for the Internet Explorer browser that may improve its behaviour with aXes.

Cache Control

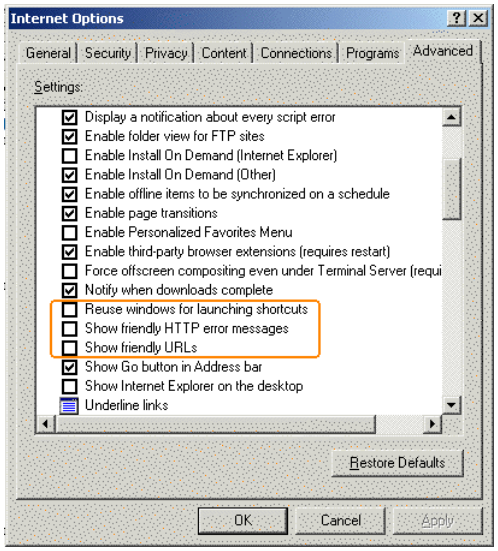
Ensure the browser is set to automatically check for newer versions of files. This option is found under the Tools->Internet Options panel on the General tab in the settings for the Temporary Internet files section.



Advanced Settings

Browsing

These settings are found under the Tools->Options panel on the Advanced tab in the Browsing section. Uncheck the “Reuse windows for launching shortcuts”, uncheck “Show friendly HTTP error messages”, and uncheck “Show friendly URLs”.

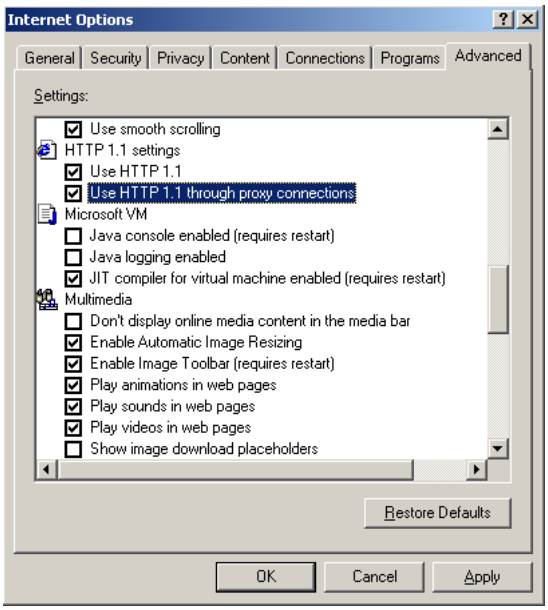


Content © LANSA Group
All Rights Reserved

Use HTTP 1.1 protocol

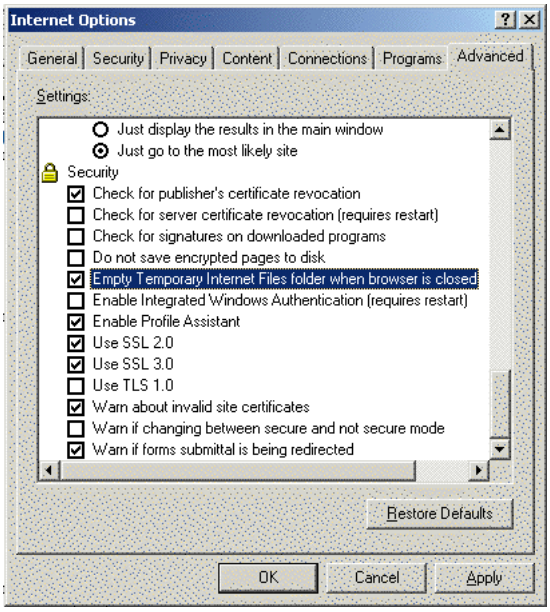
Set your browser to use the HTTP 1.1 protocol.

These options are found under the Tools->Options panel on the Advanced tab in the HTTP 1.1 settings section. Check both the “Use HTTP 1.1” and “Use HTTP 1.1 through proxy connections” settings.



Security

This setting is found under the Tools->Options panel on the Advanced tab in the Security section. Check the “Empty Temporary Internet Files folder when browser is closed” setting.



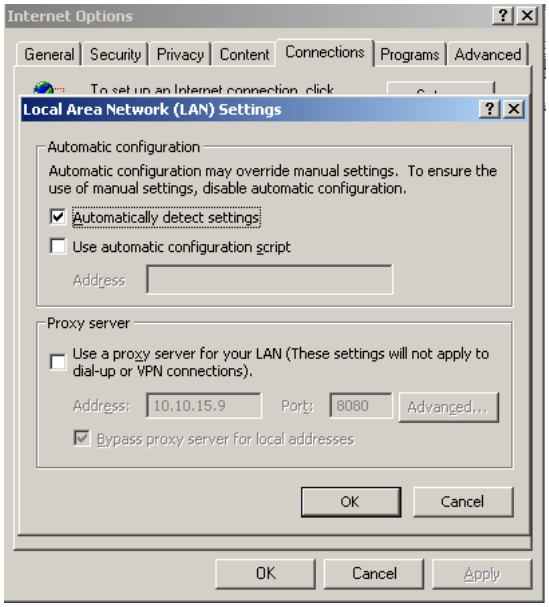
Content © LANSA Group
All Rights Reserved

Proxy Servers

Many proxy servers fail to comply with the HTTP specifications and can cause aXes Terminal Server to fail to load all the terminal files.

If you have Internet Explorer set to automatically detect settings or use a proxy server for your LAN is checked then your network administrator must ensure that any proxy server that may be found is correctly configured to pass HTTP/1.1 traffic to the server on which aXes Terminal Server is running.

These settings are found under the Tools->Options panel on the Connections tab.



If you cannot successfully load a terminal session and either of these values is ticked then you should try again after removing the tick and completely restarting the browser. If the terminal successfully loads then the problem is caused by an incorrectly configured, or non-compliant, proxy server.

Refer to the aXes Reference Troubleshooting section for information on the specific problems that can be caused by proxy servers. This section also provides a list of proxy servers that are known to cause problems either because they are non-compliant, or because they have been configured incorrectly.

Content © LANSA Group
All Rights Reserved