

What's New

aXes Version 4.1.1

Contents

- About What's New in aXes version 4.1.1 2
- Enhancements..... 3
 - New features 3
 - AutoGUI+ 3
 - New eXtensions..... 6
 - New IBM i services eXtensions:..... 6
 - New common eXtensions 8
- Developer tools 9
 - Developer tools enhancements 9
- Compatibility considerations 10
 - Proportional fonts vs Monospace fonts..... 10
 - Terminal styles. 10
 - AutoGUI vs AutoGUI+..... 11

About What's New in aXes version 4.1.1

Enhancements	New features, new eXtensions, updated developer tools, browser support and utilities.
Hotfixes	All consolidated hotfixes since the general availability of aXes version 3.1.1
IBM i server	Requires IBM i 6.1.x or higher.
Licensing	All customers installing aXes 4.1.1 will require relicensing.

Enhancements

New features

AutoGUI+

AutoGUI+ is a new feature which allows users to customize their projects by defining a set of rules. aXes will apply the rules automatically. These rules are very powerful in that they apply to each 5250 screen of your application.

Each rule is made of:

- a set of identification rules,
- one extension to apply to a target field, as defined in the identification rules,
- a set of extension properties to apply to the selected extension,
- a set of exclusion rules to avoid rules being applied to certain screen configurations.

We have also added some predefined rules to make it easier for the user to add common rules.

The extensions available for AutoGUI+ assignment are:

- AutoGUI+ menu Item
- Checkbox
- Date
- Dropdown
- Label
- Hyperlink
- Push Button
- Radio button
- Subfile scroller

There is also export/import functionality allowing developers to easily copy rules between projects or save the rules for future use.

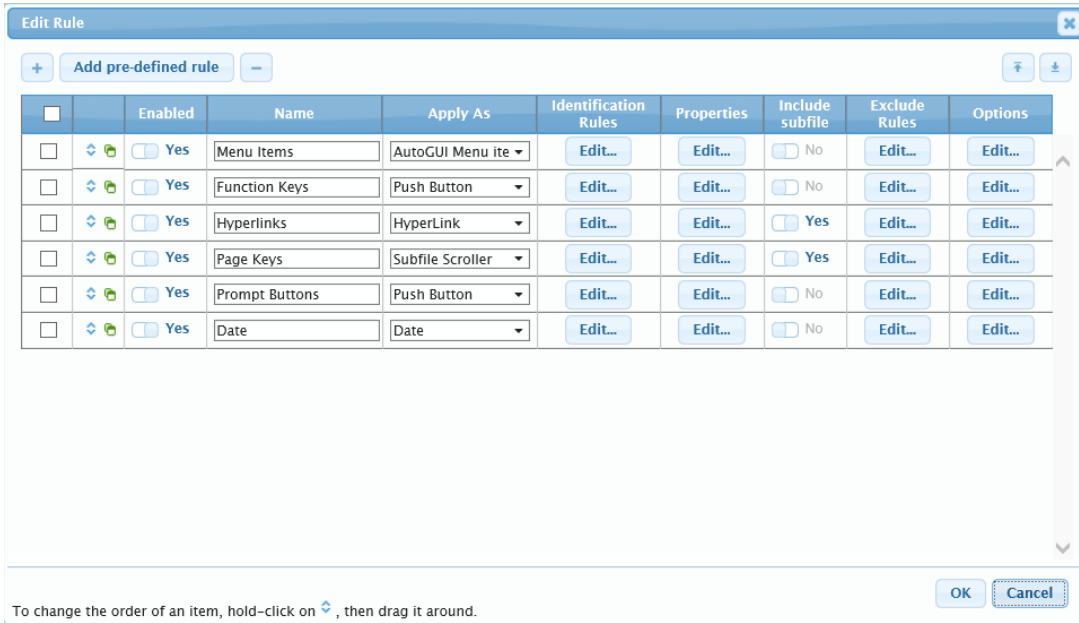


Figure 1 - Example of AutoGUI+ rules list

The identification rules allow the developer to define precisely when the rule is applied. It defines the target field, which is the field to which the selected extension should be applied, and any remarkable fields around it (left, right, top, bottom).

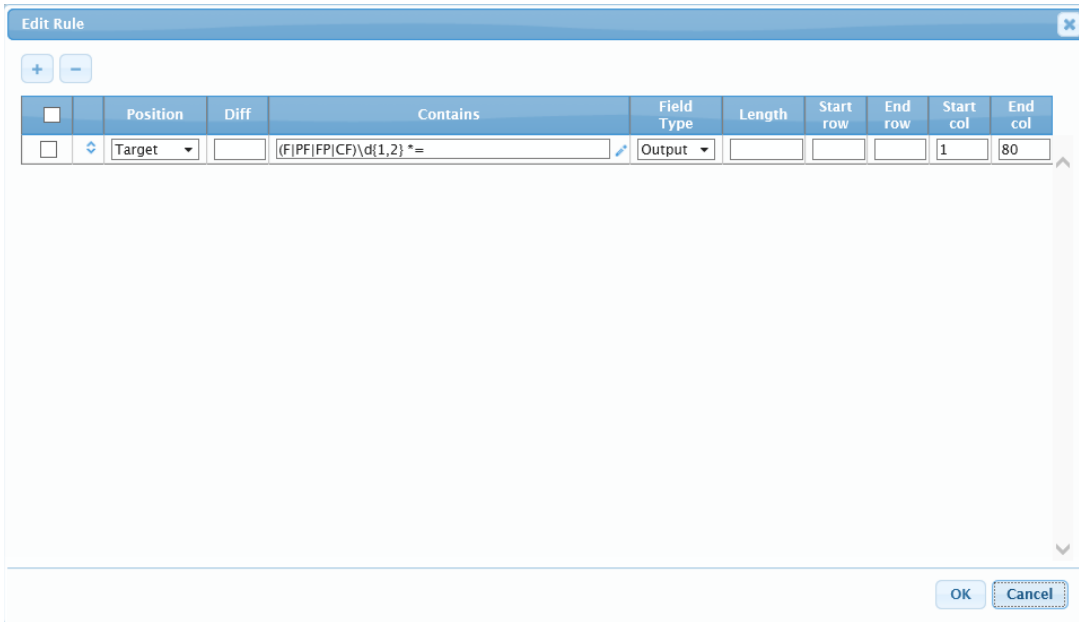


Figure 2 - Example of popup for function keys identification

It is also possible to exclude any screens or screen element from the AutoGUI+ rules by defining an exclude rule. It can be via a screen name or by identification of a unit screen element.

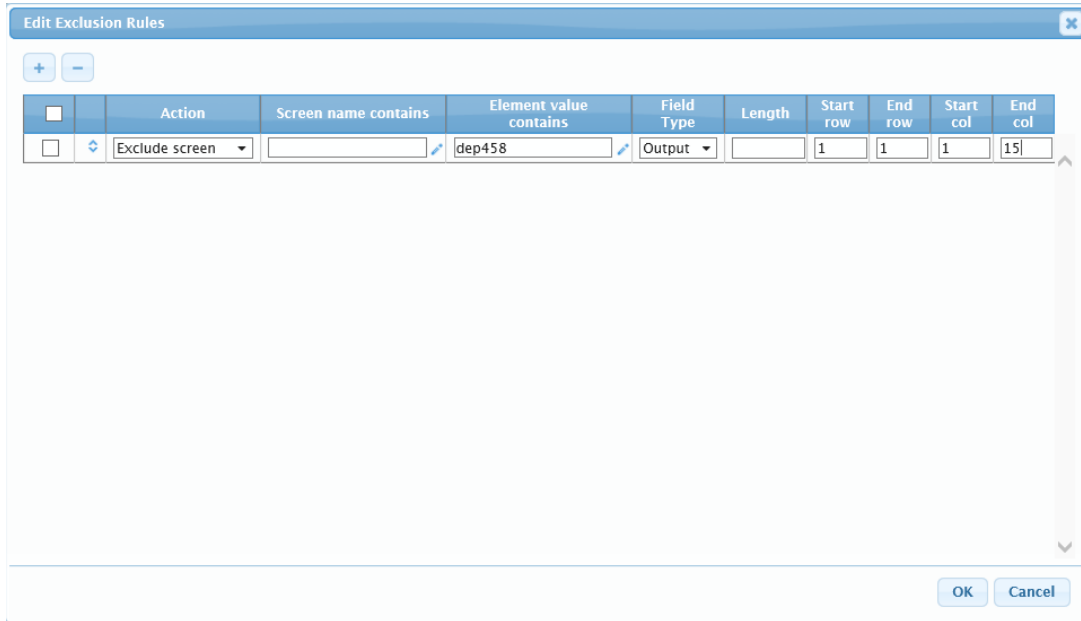


Figure 3 - - Exclusion screen example

A regular expression is a special text string for describing a search pattern. We have provided a simple regular expression generator to help in producing regular expressions from logic blocks, complete with a regular expression tester.

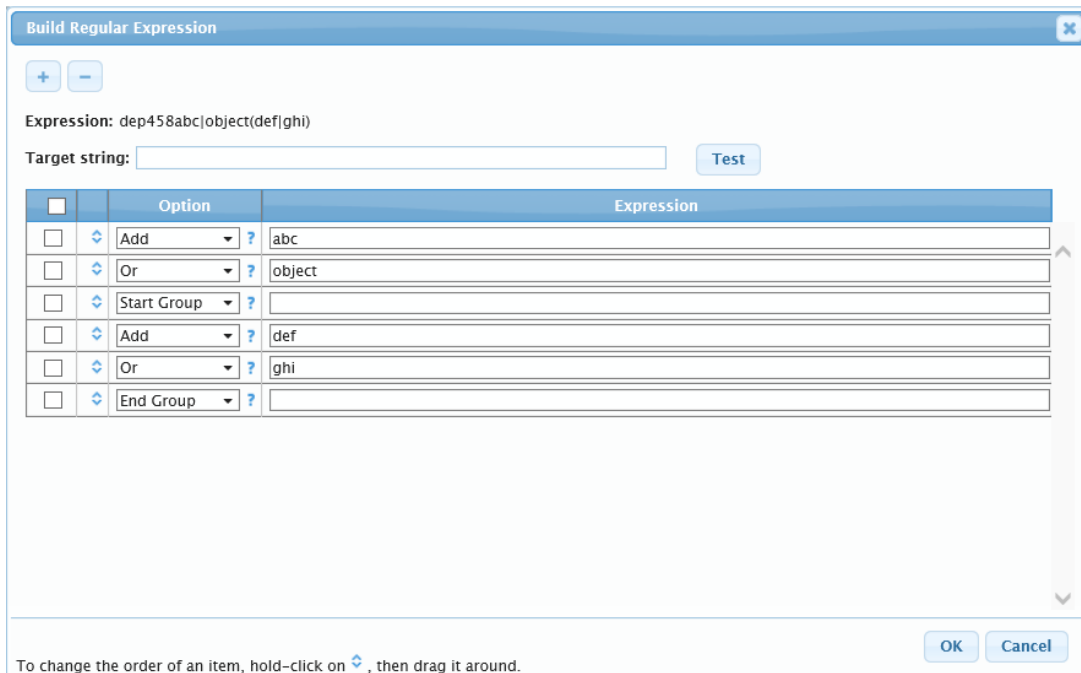


Figure 4- Regular expression generator

New eXtensions

New IBM i services eXtensions:

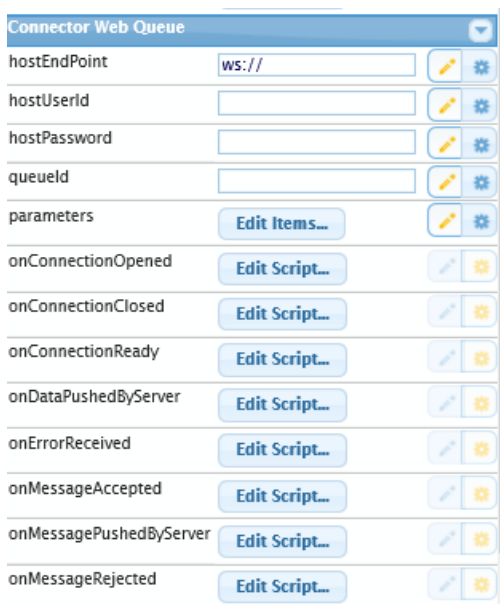
The suite of IBM i services extensions has been designed to work with aXes to provide advanced IBM i functionality to increase business value of your applications.

WebQueue

WebQueue is an API that provides communication between a web application and a server via a WebSocket connection.

Using WebQueue, developers can send/receive messages to/from program data queues using JavaScript via a WebSocket connection. Programs monitoring the program data queues can access resources and services on an IBM i server.

Developers use JavaScript and other development tools when building web applications. At points in the applications where developers require server resources, they can use the WebQueue API to access the resources. WebQueue provides web applications with a full-duplex connection to a server, over which the application and the server can exchange messages.



ConnectorJS

The WebQueue API is service provided by a feature called ConnectorJS that comes with the browser accessible ConnectorJS JavaScript library that interfaces with ConnectorJS Server to access requested server resources. ConnectorJS Server invokes the resources to produce the required results, and returns responses to web applications.

ConnectorJS also equips developers with interfaces they can use from JavaScript to access server resources in web applications. Server resources include programs, commands, files and folders, email, and messaging using program data queues and/or the Java Message Service (JMS) API.

The properties can be set via the developer tools (show below) or programmatically through the aXes API. See below for the properties available to the developer.

ConnectorJS Email

The ConnectorJS email API allows developers to assemble and send an email.

To use the API, developers code the API interface using JavaScript and deploy the ConnectorJS JavaScript library with the web application JavaScript. This API requires data defining server resources (e.g. a command, program, or data queue) and an IBM i server (host).

Email content includes subject, addressee (to), sender (from) and email body.

Developers can assemble all email details required to create generic email messages. Use the generic assembly method for emails with various subjects sent to recipients chosen when preparing an email. Inserting some of the email details in the configuration file allows CJS to use details provided via the API call and configuration details to assemble an email. This assembly method is useful for pre-prepared or specific subject matter emails such as support requests or sales information. Most of the email details can reside in the configuration file and developers can provide the email body content via a property in the API call.

To:	<input type="text" value="TO Recipients here"/>
Cc:	<input type="text" value="CC Recipients here"/>
Bcc:	<input type="text" value="BCC Recipients here"/>
Subject:	<input type="text" value="Subject here"/>
<input type="text" value="Message here"/>	
<input type="button" value="Clear"/> <input type="button" value="Send"/>	

ConnectorJS Program Call

The ConnectorJS program call API allows developers to call a program.

The API call can return a single value once the call has been processed. The properties can be set via the developer tools (show below) or programmatically through the AXES API.

See below for the properties available to the developer.

Connector Program Call		
hostEndPoint	<input type="text" value="http://"/>	
hostUserId	<input type="text"/>	
hostPassword	<input type="text"/>	
program	<input type="text"/>	
parameters	<input type="button" value="Edit Items..."/>	
asynchronous	<input type="checkbox"/>	
onErrorReceived	<input type="button" value="Edit Script..."/>	
onPgmCallCompleted	<input type="button" value="Edit Script..."/>	

ConnectorJS Print Service

Print a report from a browser using the ConnectorJS Print Service (CJS)

The CJS Print Service enables developers to prepare a report and send it to a printer accessible from a browser. The Print Service collates data formatted as JSON with a report layout and content specification to produce the report. It can accept data from spooled files, data from CJS Query results, data from an aXes terminal screen, or data generated by JavaScript.

What can the print service do? Suppose you want to take invoice details from an IBM i spooled file and print an invoice, adding details that do not appear in the spooled file, e.g. a company logo. Another example is printing a receipt for an online order.

Developers create a report layout by defining page elements including text, images, and graphics, and their page location. Page properties include, title, DPI (page resolution), size (e.g. A4), orientation (portrait and landscape). Each element (e.g. text, bar code, line, rectangle, or image) on the page has a set of properties defining how to present the element. The properties include location on the page (x and y coordinates), font, colour, line properties (e.g. style and width), rotation, and URL representing an image location.

New common eXtensions

New common extensions are available for desktop and mobile, and only for TS2.

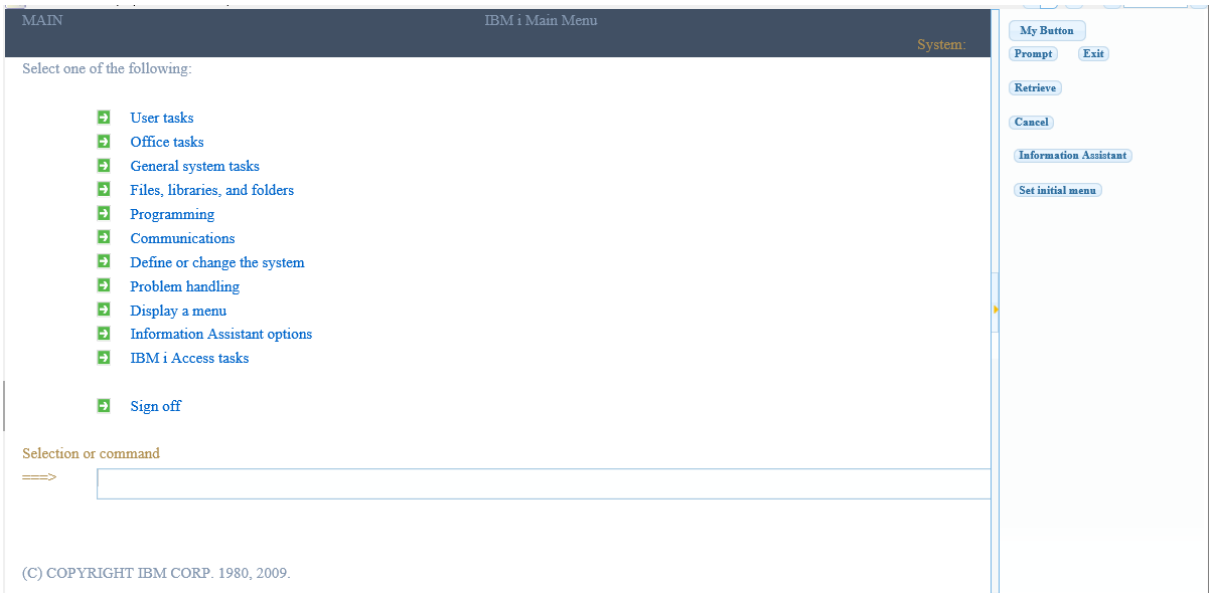
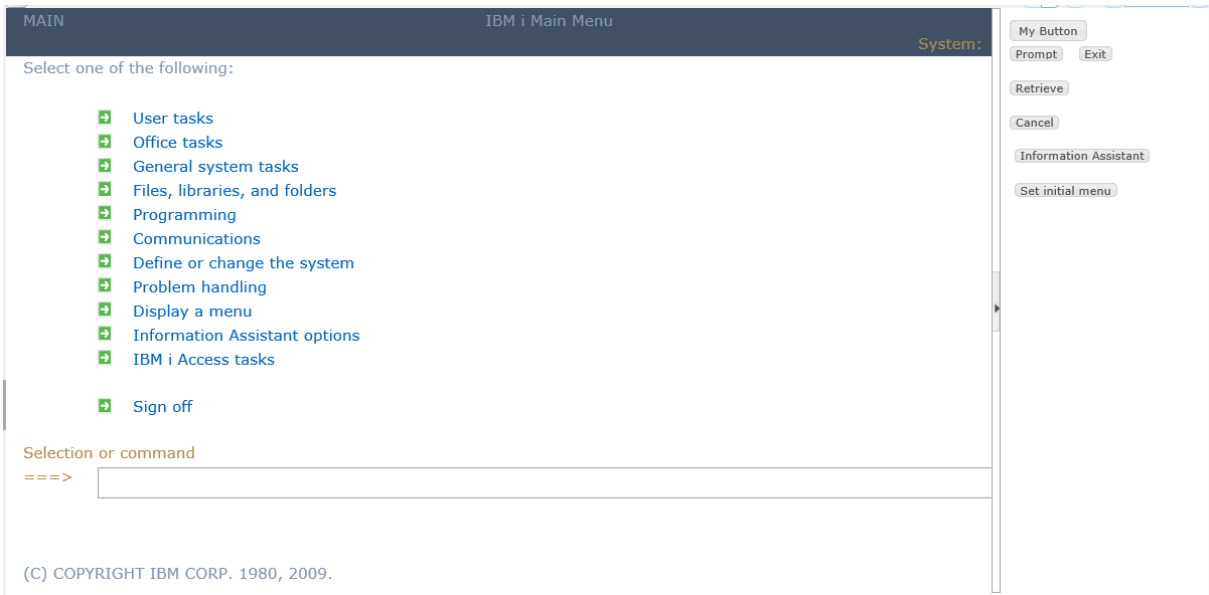
Multiline Edit Box

The Multiline Edit Box allows user input of more than one line of text. The text display depends on the size of this extension. The text will go to the next line when it overflows the width while vertical scrollbar becomes visible when the text overflows the height

Button Panel eXtension

The button panel eXtension allows users to insert a panel on the screen where buttons can be added. The panel is a container which can be positioned Top, Right, Bottom, Left of the screen, where buttons can be dragged and drop into.

Customisation of individual buttons is supported, any newly button or existing screen button such as Function keys button can be dragged and dropped in to the panel.



Developer tools

Developer tools enhancements

TS2 development tool modernization

The TS2 development tool has been enhanced with the following features.

- The editor has been updated to apply the jQuery UI theme throughout the User Interface in a consistent manner, providing the developer with a unified look and feel.
- The theme of the developer tools is customisable through the configuration file.

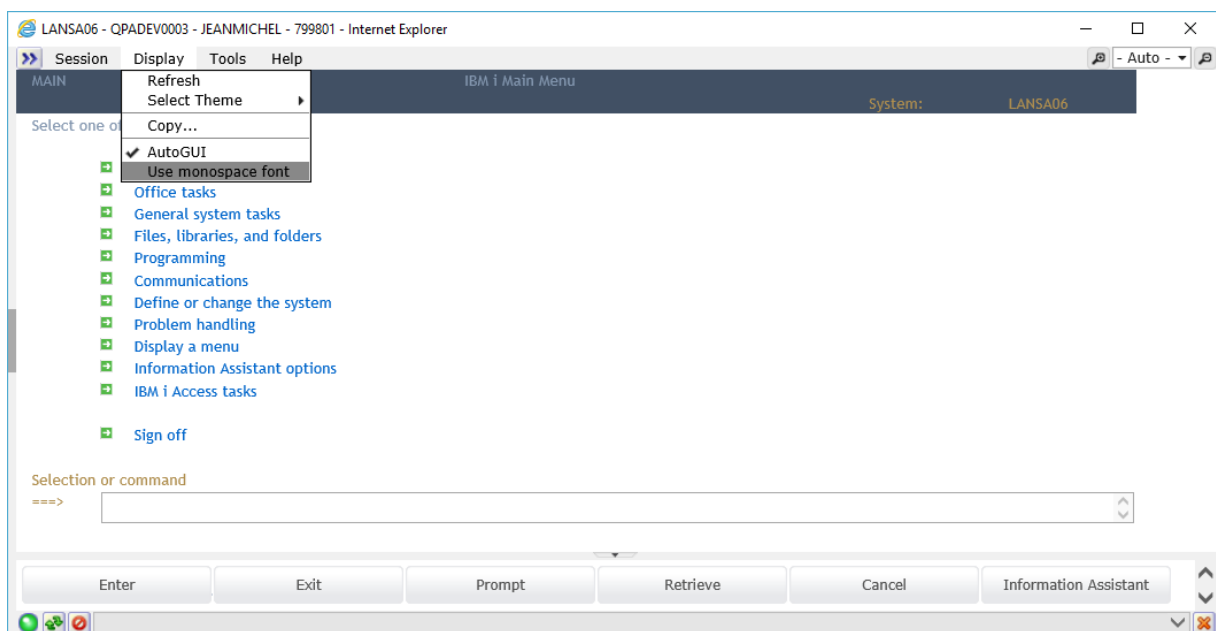
Compatibility considerations

With this release, we have introduced many performance and visual improvements. In this respect, the user is advised to read the following point which explains changes which might impact their application.

Proportional fonts vs Monospace fonts.

aXes 4.1.0 has introduced proportional fonts thus screens might look somewhat different and misaligned in some circumstances. To help with these changes we have introduced an option on the display menu of the terminal window which allows the user to switch easily from proportional to monospace font.

See below for an example.



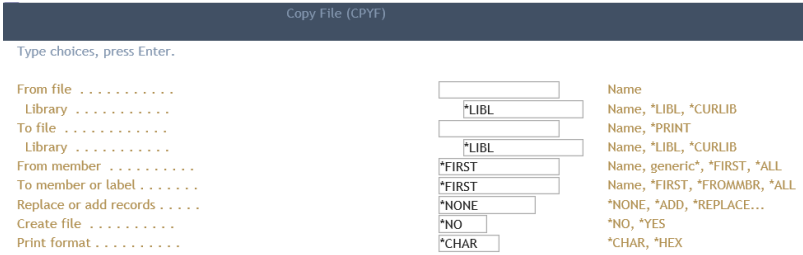
Terminal styles.

The styles throughout the product utilise the jQuery themes more consistently. Consequently, some screens might have a different visual appearance since the old CSS themes are being retired.

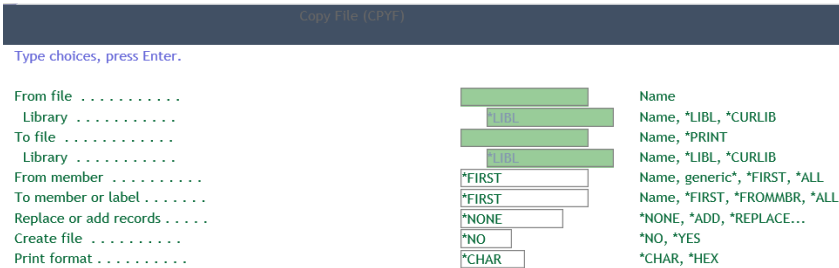
One solution is to use the new AutoGUI+ to apply the terminal styles to your application.

You can add them by adding a predefined rule in AutoGUI+ called Terminal Themes.

The screen below shows the Copy File screen with jQuery Themes only.



The screen below shows the Copy File screen with jQuery Themes and Terminal Styles active.





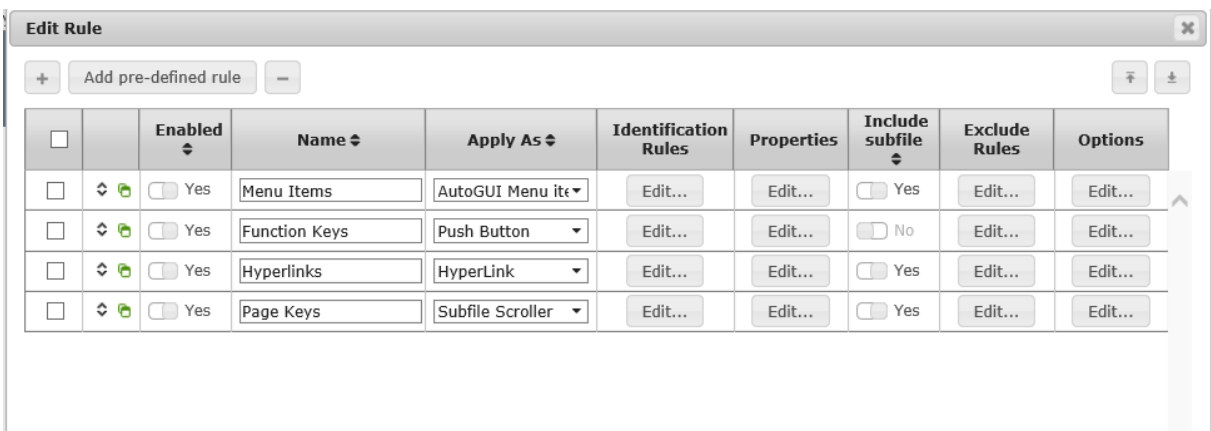
AutoGUI vs AutoGUI+.

AutoGUI refers to the legacy automatic refactoring feature of previous aXes releases. AutoGUI+ is the new feature released with this version.

AutoGUI+ is an upgrade from AutoGUI rules. In some cases, your existing projects might implement upgraded AutoGUI+ rules. This could be the case if you have never changed the default AutoGUI rules in your project, therefore never saved any version of your rules.

To help user who wish to keep the existing AutoGUI behaviour we have added an import of the basic AutoGUI rules.

To revert to the legacy rules, open the AutoGUI+ window, check all the rules displayed and click . After confirmation, you have an empty list. In the same popup click on import  and select the file /ts/screens/defaultautoguirules.json. After validating the import, the user will have reverted to the legacy set of rules.



Hotfixes

Consolidated hotfixes

Since aXes version 3.1.1 a number of new features and corrections have been issued as hotfixes. These hotfixes have been consolidated into aXes version 4.1.1.

The full list of hotfixes is available from the aXes Forum: <http://axes.lansa.com.au/>