

Advanced Operations with Co-Pilot for z/OS



Overview

Co-Pilot is an automation tool designed to increase z/OS system application and resource availability. Co-Pilot improves operations to achieve a 24/7 automated environment.

Co-Pilot can be configured to monitor and respond to system events in a z/OS system.

Co-Pilot can be configured to schedule tasks at a desired date/time interval.

Co-Pilot can control the display characteristics of console messages.

Co-Pilot can be configured to monitor critical system applications.

Co-Pilot is based on a TABLE concept, allowing the user to create rules based on a group or category. The user may create a set of rules, and then implement the rules when the time is appropriate.

There are 2 sets of datasets that Co-Pilot uses to maintain the automation environment. There is a working set of data sets, that are used to build the automation rules, and there is a run time set of libraries, that are used to process and monitor events in a z/OS system. The user creates and builds rules in the working set of libraries using the ISPF interface. When ready for implementation, a load command is issued to dynamically load the table.

Co-Pilot Rules

Co-Pilot can control the following rules for automation:

- **Message Action Rules** – This rule type traps messages by the message ID, and issues a command when triggered.
- **Time of Day Rules** – This rule type triggers a command based on the time of day rules specified.
- **Monitor Rules** – This rule type allows for monitoring critical address spaces status. It also allows storing of start and stop commands.
- **Message Display Rules** – This rule type enables modifying a message's display attributes.

Co-Pilot Objects

- **Programs** – There are 2 types of programs that Co-Pilot controls.
 - **Co-Pilot Program** – This type of program contains a set of in-stream Co-Pilot commands. Co-Pilot programs run in the Co-Pilot Master address space.
 - **REXX Program** – This type of program is a standard TSO/REXX program. REXX programs run in a Co-Pilot TSO processor address space.
- **Global Variables** – Variables can be defined as follows:
 - **Static** - Variables whose values do not change from the time they are loaded.
 - **CKPT** - Variables whose values can change from the time they are loaded.
 - **NOCKPT** – Variables that can change but only persist until Co-Pilot is recycled.
- **Holiday Calendar** – Specify holidays that Co-Pilot will recognize as holiday dates.

Co-Pilot Run-time Environment

The Co-Pilot run-time environment consists of 3 components.

- **Co-Pilot Master** – This is the main server for controlling the rules and monitoring events within the z/OS system.
- **Co-Pilot Subsystem** – This is a subsystem that traps messages that have been marked for changing the display attributes. It also allows for a character substitution of the Co-Pilot MODIFY command.
- **Co-Pilot TSO Processors** – This is a set of TSO processor address spaces to execute TSO commands and REXX programs. The number of TSO processors is determined by the system startup options specified.

Co-Pilot ISPF Interface

The Co-Pilot ISPF Interface is a series of ISPF dialogs, to build and maintain the Co-Pilot automation rules. The table-based system allows the system administrator to build the automation rules that are based on categories and/or groups.

The Co-Pilot ISPF interface is a work area for the user to build and maintain the Co-Pilot automation rules. When making changes to one or more entities in the Co-Pilot system, a LOAD command is required to implement the changes. This allows the user to activate the changes only when they are ready.

Call Toll-Free 1-888-873-8273 (USA & Canada) for more information or visit our website www.cartagena.com

Cartagena® Software Limited

101 Drawbridge Drive, Markham, Ontario, Canada L6C 2N5

Tel: +1 905-887-0755 E-mail: info@cartagena.com

Cartagena is a registered trademark of Cartagena Software Limited.

