Introduction

The Master Console Center (MCC) automates and enhances data center operations by:

- Providing a high degree of operational consolidation, allowing multiple data centers and/or hosts to be monitored and controlled from a single location.
- Providing monitoring of mainframes, mid-range (IBM I Series/ AS400) and open systems hosts.
- Monitoring host operations, including scanning of messages and signals, and responding to operational processing.
- A command scripting language, GCL (Global Command Language), commonly used by operations personnel to customize automation.
- Issuing alerts associated with all processes performed or observed.
- Initiating recovery responses.
- Starting and stopping processes, including system startup and boot activity (for example, IML, IPL), and system shutdown (for example, OS shutdown).
- Complete AAA security and optional integration with RACF or other LDAP servers.
- Complete auditing and console history to help pass the most stringent Sarbanes-Oxley requirements.
- 100% remote maintenance capabilities accessible via a dial-up modem or via ssh or telnet.
- All MCC features are available via a character console allowing access to consoles over slow speed communications lines.

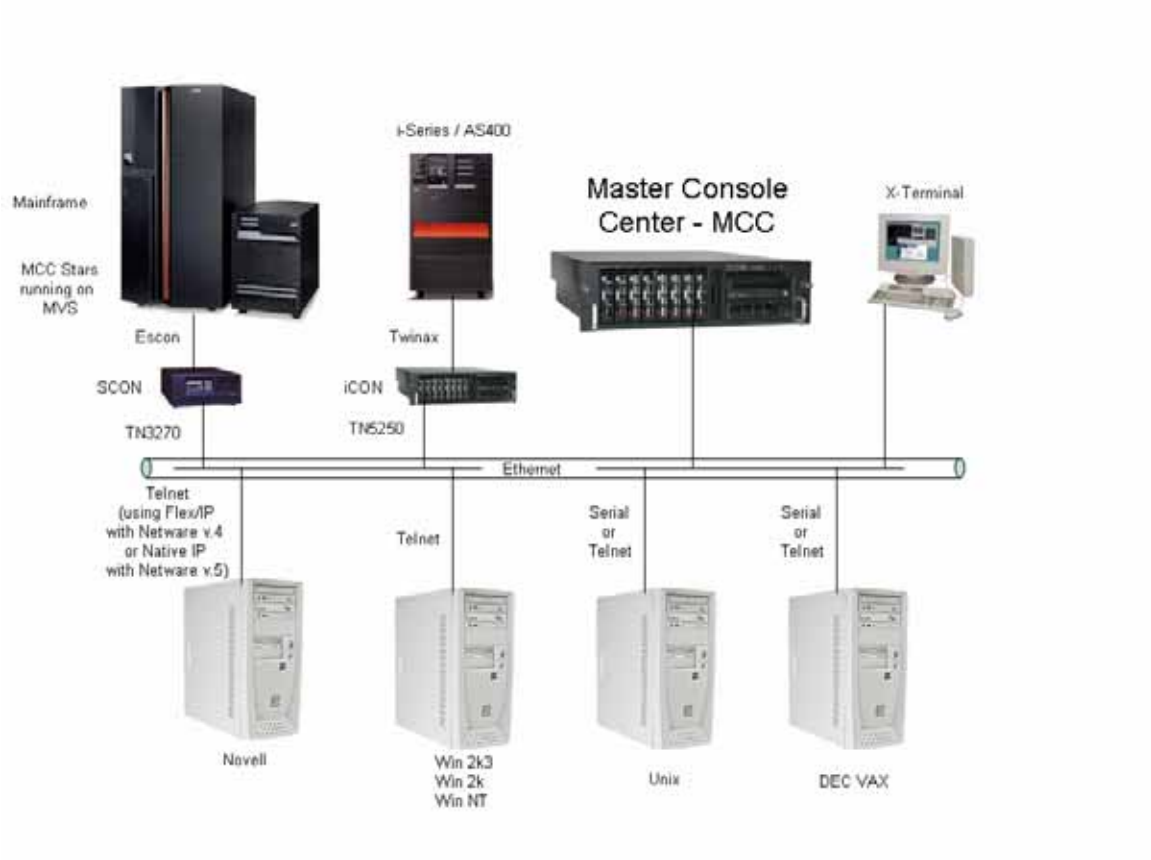
The MCC is delivered in one of the following configurations:

- As a hardware/software package, comprising both a server and the MCC software. If access is required to systems such as those that cannot be reached from the network, Visara can supply other hardware kits to provide the necessary interfaces as part of the package.
- As a complete turnkey solution. In this scenario, Visara is responsible for providing and installing all hardware and software and bringing the system on-line. The customer is responsible for site preparation and certain other prerequisites such as assigning IP addresses for the equipment.

Note: All MCC formats have the same general capabilities, and can use the MCC software to monitor mainframes and server systems. The main advantage of the turnkey solution is that much of the preparation and installation tasks are completed by Visara personnel, thus easing the additional workload on the customer's personnel.

MCC Architecture

Here is a typical scenario in which the MCC can be used.



The MCC software is installed on a dedicated Red Hat Linux server, supplied by Visara or the customer. It can communicate with systems and devices that are connected to the network, typically using TN3270, 5250, RS-232, telnet, VNC and SNMP. This includes Open Systems, Midrange systems(i-Series, AS/400), Mainframes(s/390, z-Series), most Windows platforms and other devices.

At any time, Visara's hardware technology can be added to the installation to monitor mainframes, AS/400s, or other systems without Ethernet connectivity.

From the operator's viewpoint, all systems can be monitored from the MCC terminal, regardless of how they are connected to the MCC server.

Hardware Management Consoles (HMCs) can also be accessed using the MCC. For additional automation that provides robust alerting for mainframes, a small optional agent is available to handle high volumes of messages from mainframes.

Main MCC Uses

All MCC installations have the capabilities described below.

Operational Consolidation

The MCC GUI is available via the X Windowing System. Terminals can be located anywhere on the customer's wide area network (WAN) or local area network (LAN). From an MCC terminal, a single operator can monitor and manage distributed systems of different types anywhere on the LAN or WAN. Alternatively, several MCC terminals may be installed and monitoring responsibility can be divided between several locations or operators. The capability permits the elimination of multiple monitoring interfaces, each of which can only monitor one system or a single type of system.

Event Monitoring

The MCC can accept input events in many different forms, including:

- Console messages from mainframe or server systems.
- SNMP traps from any variety of networked SNMP agents. The MCC can also get and set SNMP MIB variables.
- Command line events. Through the MCC's generic event handling feature, a command-line program can send arbitrary events to the MCC.
- Output of any command typed at a console. MCC automation scripts can type any command into any system. The responses to these commands are captured by the MCC.
- Time messages scheduled by the MCC Event Manager or scripts written in GCL (Global Command Language).

Event messages are typically generated in response to system events such as power failure or restoration, unauthorized logon attempts, filled disks, and so on. Messages may be filtered and the level of notification configured according to their severity.

Event Response and Notification

The MCC can output commands and/or messages in response to events in several forms, including:

- User-customizable graphical displays on the MCC terminal. MCC alerts and messages appear on the MCC GUI.
- Direct command(s) to an MCC managed system. The MCC can enter commands into any console connected to the MCC. This allows the MCC to act as if the operator typed in these commands on a system connected to the data center.
- Other Linux programs. Since the MCC executes on a Linux server, any Linux program can appear on the MCC interface.
- Other systems management packages. The MCC can interact with most packages that support SNMP.
- E-mail. The MCC can send SMTP e-mail.
- Pages. The MCC can send messages to certain paging systems.
- Signboard message.

Automation of Responses

The MCC can be programmed in Visara's Global Command Language (GCL), allowing the user to customize how the MCC responds to certain console messages and conditions, including:

- Errors
- SNMP traps
- Operator-set thresholds

The MCC's Event Manager can also schedule the execution of GCL scripts as a monitoring or preventive maintenance tool.

SNMP

SNMP stands for Simple Network Management Protocol. Essentially, this is a way for equipment to volunteer status information over the network.

The MCC is capable of both receiving and generating traps. In addition, the MCC interact with any fully-SNMP compliant host, allowing more advanced monitoring.

Monitoring Systems with the MCC

The main user interface on the MCC terminal comprises a number of GUI windows for monitoring systems and configuring connections to those systems. The most commonly used capabilities provided for monitoring systems by each GUI window are described below:

- **Alerts**. The alert window displays alert messages, and their status (new, open, or closed). The administrator defines the system conditions that generate alerts; the MCC's responses to these conditions are then defined using the Event Editor window. Alerts may be created, modified, or deleted by users and by scripts.
- **Console Selection**. The Console Selection window allows the user to select and use the consoles that are connected to the MCC, view their status, and lock or unlock them.
- **Master window**. Used by the administrator to view active users, log out of, and administer the MCC.
- **Security Editor window**. Allows security groups to be defined, users assigned to those groups, and login requirements defined.
- **Event Editor window**. Defines how events are handled by the MCC, including the notification(s) supplied and any script that is invoked if an event occurs.
- **Script Editor window**. Allows Global Command Language (GCL) scripts to be written or modified. GCL scripts automate the monitoring of systems and responses to events.

Using the MCC

To make effective use of the MCC, the customer must identify and train essential personnel. The following personnel are required:

- **System Administrator**. The system administrator must become thoroughly familiar with the MCC, and is responsible for activities such as:
  - Assigning and creating login IDs.
  - Starting up the MCC system.
  - Shutting down the MCC.
  - Backing up the system.
  - Making configuration changes.
  - Testing and implementing scripts.
  - The system administrator should also be familiar with the Linux environment, and X-windows functionality.

- **Operator**. The operator is responsible for the day-to-day use of the MCC, and should be familiar with the MCC terminal and the GUI windows. This person should be conversant with the applicable systems connected to the MCC, the implications of possible events, and the actions to take in response to alerts.
- **Script Programmer**. The MCC can provide a high degree of automation of the response to events. This automation is implemented by the use of scripts written in GCL. If a site intends to make extensive use of automation, an operator or programmer should be trained to write and maintain the necessary scripts.

Preferred Connection Methods

The MCC is capable of connecting to a wide range of external systems using the interfaces outlined previously. *Table 1 MCC Connectivity Chart* provides a summary of the systems to which the MCC can connect.

System	How to connect to MCC	Notes
Mainframe	TN3270	Requires Visara SCON platform or OSA
IBM Shark Storage Controllers	IBM 3151 terminals	
AS/400	TN5250	Requires Visara iCON platform
Unix	RS232 console, Telnet	
HP3000 HP9000	Telnet, HP700/92	VT100 Telnet may also be used with some applications, but may not work correctly with administration programs
Tandem	RS232 console, Telnet	RS232 requires Tandem AWAN server
VAX/VMS	RS232	
Novell NetWare	Telnet, RS232	May require an IP package installed on the server
Windows Server 2003/2000/NT Windows Desktop Vista/XP/NT	Telnet, VNC, RS232	May require additional Telnet service that is not supplied with standard Windows OS
Other (switches, routers, remote power switches), etc.)	Telnet, RS232	RS232 may require additional hardware

Configuration Requirements

After installation of the MCC software, it must be configured to work with each system. Typical tasks include:

- Configure hardware. If the interface hardware was installed, it must be configured for correct operation with the customer's systems.
- Create telnet sessions. Define a telnet connection for each system to which the MCC will telnet.
- Create MVS sessions. Define an MVS connection for each MVS system to which the MCC will connect.
- Define the systems to be monitored by adding Rooms, Groups, CPUs, and OSs.
- Create additional users and assign security. Create additional user accounts for each MCC user on the server on which the MCC is installed. Create security groups, and assign users to relevant groups.
- Define Events and define how the MCC responds to each event.

Refer to the M*CC Administration Guide* for a complete list of requirements and procedures.

MCCSTARS MVS Agent

Overview

The Visara MCCSTARS MVS agent is designed to cope with MVS's message traffic and offload most filtering from the MCC server. MVS LPARs typically limit operator messages so that operations personnel don't see many messages. Unfortunately, in critical situations MVS LPARs can create hundred or thousands of messages a second. When managing 10s to 100s of LPARs it becomes imperative to intercept console messages at the source thereby offloading processing of screen scraping and filtering from the MCC server. Competitive products do not operate in this manor and typically do screen scraping at the server level, which requires the most powerful servers and typically still have slow-downs during message storms. In addition, competitive products often require consoles to be placed in "Roll Deletable" mode to quickly detect when action messages are deleted. If messages on these consoles back up, MVS may experience a WTO buffer shortage and, in extreme cases, shut down.

MCCSTARS is designed for high traffic. Out of the box it only passes messages flagged as important by the issuing program to the MCC server. There, user customizable MCC GCL scripts can further refine processing. Again, out of the box messages and WTORs are used to create MCC alerts. As operations personnel resolve problems and perform necessary actions, MVS software indicates that messages are to be deleted. MCCSTARS passes this information to MCC and deletes the associated alerts. In high traffic situations, MCCSTARS meters out important messages to limit network and VCC traffic.

MCCSTARS uses industry standard SNMP for its network communication and only IBM approved APIs for its MVS interface. Thus, it is protected from MVS changes and installs in under an hour.

Customization

There are two optional methods of customizing MCCSTARS operation:

- MCCSTARS GCL user exit script
- MVS resident filtering facility

User Exit Script

The optional, MCCSTARS GCL user exit script can be customized by operations personnel to ignore messages, change message text, change the alert color, route messages to specific classes of operators and send SNMP traps to other software systems.

Many of customers route tape messages to tape operators only. If tapes are not mounted within a site-determined time limit, the alerts are escalated to more senior operators.

Another use of the exit is to pass alerts to paging software via SNMP traps or to send email.

MVS Resident Filtering Facility

Optionally, MCCSTARS will also filter messages on MVS, before determining whether to ignore them or send them on to the MCC server.

For example, MVS creates a message when jobs end. In almost all cases this message can be ignored by operations. When a critical job such as CICS, which supports thousands of online users ends, MCSTARS can search for the CICS ended message on production LPARs and cause an MCC alert.

Some programs create messages that, while flagged as requiring operator action, do not. For example, a remote printing product flags printer out of paper messages as requiring operator action. Remote users, not operations, normally handle this condition. Using MCCSTARS filters, these messages can be ignored on the LPAR. Alternatively, these messages can be ignored at the MCC server using the GCL user exit script.

Installation

MCCSTARS runs as an MVS-started task. It can be installed in under an hour without an IPL. Three MVS data sets must be created, an MVS command must be issued and a simple JCL member must be added to an MVS procedure library. MCCSTARS can be run as a batch job during initial testing and then changed to a started task at customer convenience.