DANEXiS
NOSERGROUP

Media Device Control

DANEXiS AG
Bern | Winterthur
Galgenfeldweg 18
CH-3006 Bern
+41 31 33 99 888

www.danexis.com

Overview

MDC (Media Device Control) is a control and automation framework for media and broadcasting companies. The MDC framework includes Server, Admin and Panel components that cover the entire range of control requirements with a single product.

Benefits

The technical complexity of studio and broadcasting infrastructure has grown enormously in recent years and will continue to grow. With MDC, your multilayered studio and broadcasting infrastructure can be easily and seamlessly managed for the user. Complex control and automation tasks can be abstracted.

With MDC you can

Control your transmission chain:
– Studio switching
– Regionalisation
– Multichannel operation
– Backup switches

Automate your studios:
– User personalisation
– Signalisation
– Presets

MDC communicates with many devices from the media sector, including:

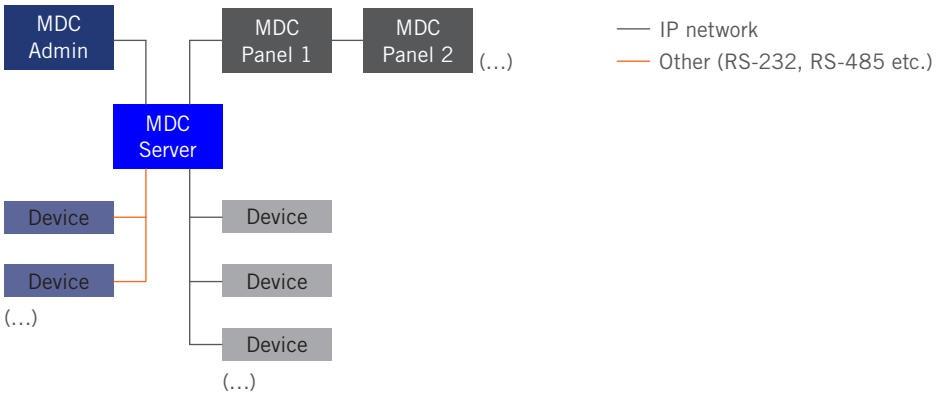
– Audio and video mixer
– Audio and video routers
– DSPs
– Audio and video analysers
– DMX lighting systems
– Desks
– Visual radio
– Cameras
– Ember(+) devices
– ProBel devices

To achieve control at all levels, MDC comes with support for a range of auxiliary and add-on systems, such as:

– Power control
– Display control
– KVM
– Signalling systems
– Web media/export
– SNMP devices

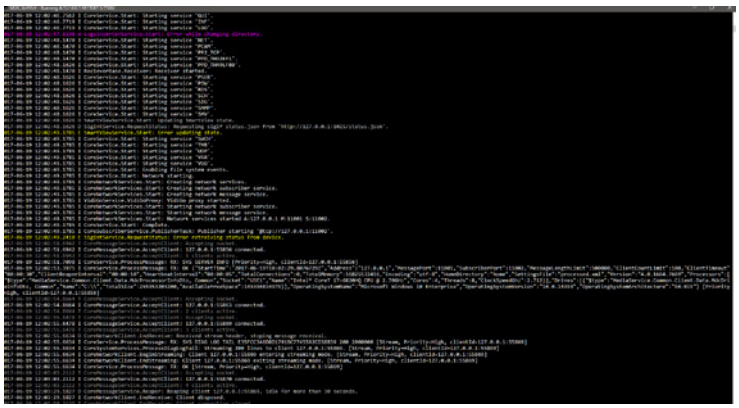
Further systems are integrated on request, quickly, easily and in a straightforward manner.

Components



Server – Admin – Panel

MDC

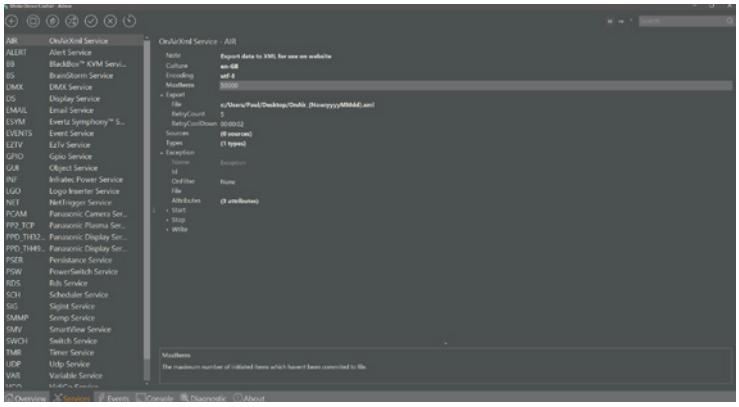


Server console

Server

MDC Server is the core of the framework. The Server controls all connected hardware and software infrastructure on a modular basis. An event engine manages the automation and control.

Admin

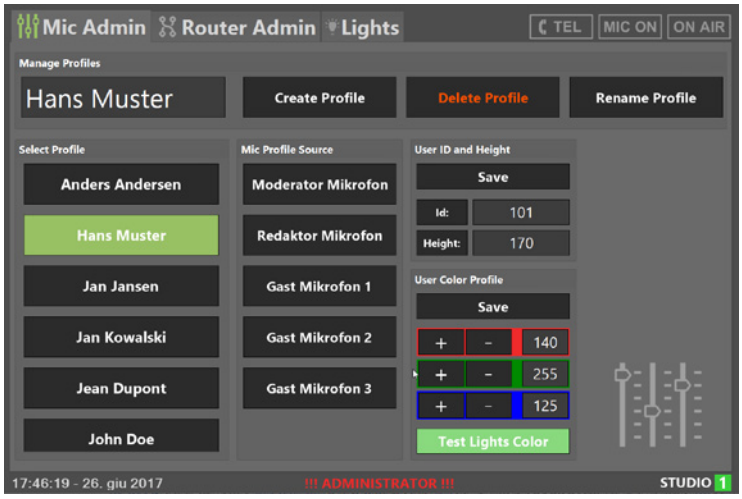


Admin user interface

Admin

The Admin UI enables comfortable configuration of the Server and provides monitoring functions.

Panel



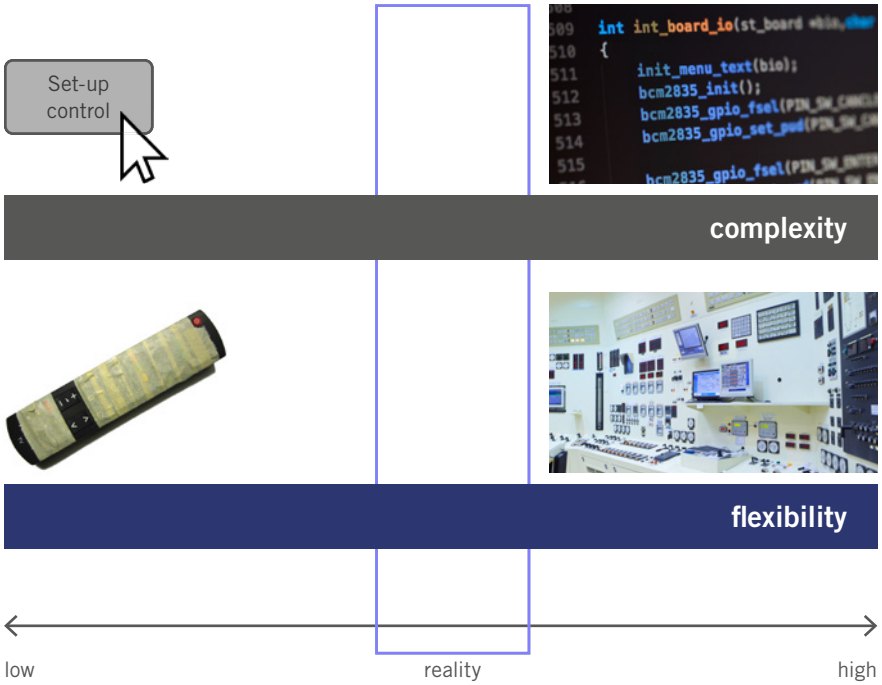
Panel interface

Panel

The Panel provides a user interface and can be adapted to your application.

The Challenge

With each automation and control solution, there is a compromise between flexibility and complexity. A highly flexible, customised solution is complex to implement, whereas an easy-to-implement solution cannot be flexibly adapted to all requirements.

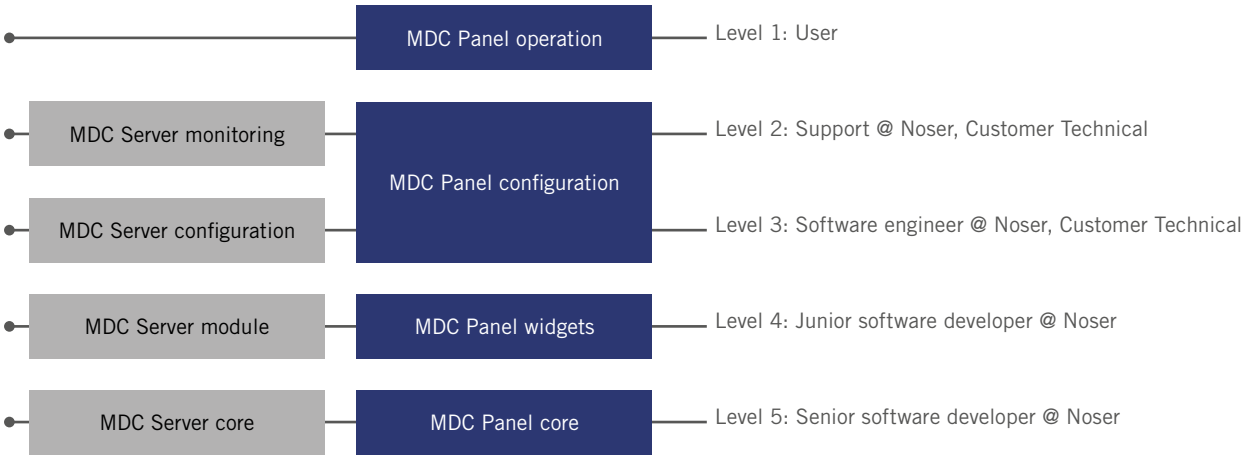


Compromise

The Solution

The challenge above cannot be completely solved, yet we strive for high flexibility and try and integrate as much of the complexity as possible.

In order to simultaneously achieve maximum flexibility and the simplest possible configuration and operability, the configuration of the MDC framework is divided into different user skill levels.

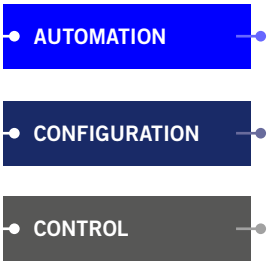


Build according to skill level

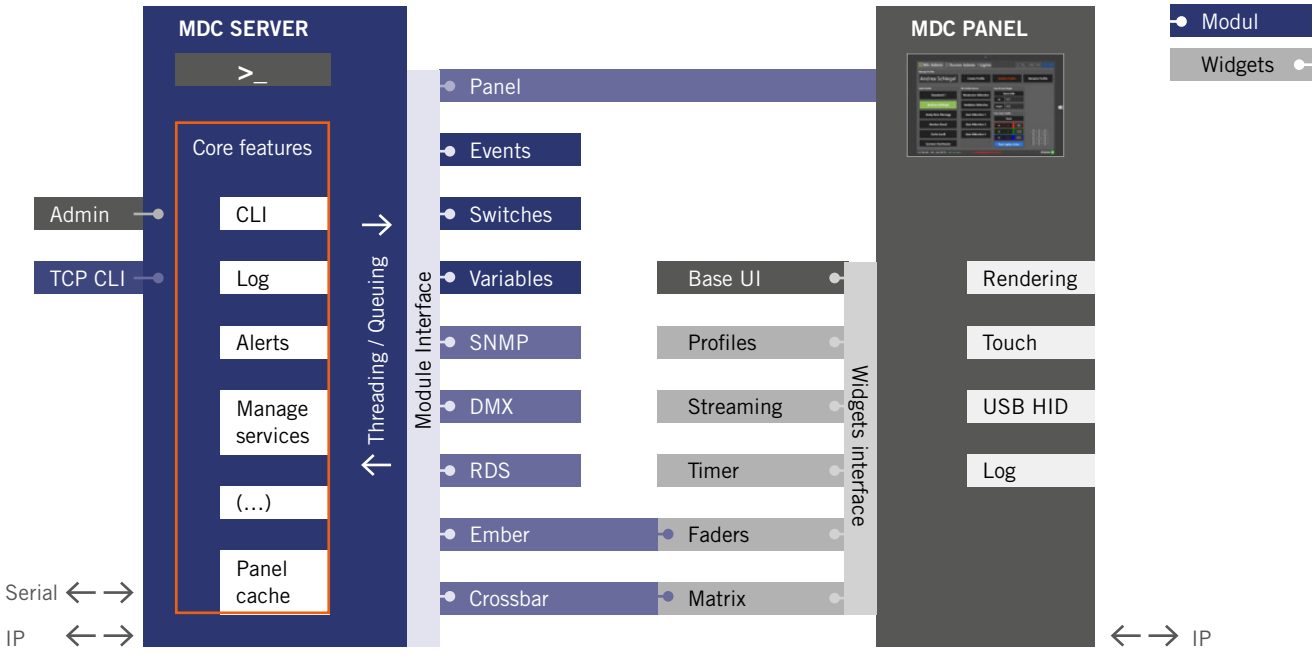
Modularity

Customer requirements for control and automation software are always very specific. In addition, the cost pressure on a control solution is usually high, since the priority for the customer is, of course, the rest of the infrastructure.

In order to take this aspect into consideration, MDC is modular, both technically and in the licensing model. A high number of existing modules and the possibility to develop new modules quickly and cost-effectivly, allow us to adapt the system as closely as possible to customer needs at a reasonable cost.



Customer requirements



MDC modularity

Server Modularity

Modularity enables adaptation to a specific infrastructure, to a large extent without the usual special developments.

For many tasks, core framework functionalities can be used without having to develop these for the specific implementation:

Logging
Comprehensive logging ensures that problems can be quickly localised. With different log levels the log can be adapted to the requirements. In addition to Server core and module log entries; the event or Panel configuration can also be configured to create log entries.

Monitoring
The Server admin user interface is used to monitor the Server. It provides access to the Server console, client console, Server log files, and statistics.

Alerts
Errors and exceptions that occur can generate alerts sent via email.

CLI
The command line interface makes it possible to access core functions, modules or events directly and is very useful when testing and configuring. The CLI uses the same command syntax as the event engine.

Functionality that goes beyond the core features is offered on a modular basis.

The basic functionality of control and automation is also connected via the module interface:

Events
Events can be triggered by modules as well as by commands. You can configure any of your own events. (See 'Event Engine' for more information.)

Switches
Switches are self-defined multistate switches. These can be set and queried via commands. A status change automatically triggers an event.

Variables
Variables are user variables that can be used arbitrarily and in a global context. As an option, each variable can be configured as persistent on request – the respective value is not lost during a restart of the MDC Server.

Panel
The Panel interface module ensures communication with the panels. It internally accesses the Panel cache, in which all runtime objects are kept.

Devices/software/special implementations
Each piece of hardware or software to be controlled is connected via a device or protocol-specific instance of a module. Currently there are more than 40 infrastructure modules available, which include industry standard protocols such as SNMP, DMX and Ember(+), as well as proprietary interfaces for various media and broadcast media products.

Because modules can benefit from the core functionalities of the Server and are developed as libraries, further customer-specific modules are very cost-efficient.

All modules and their instances can be conveniently managed and configured in MDC Admin. (See 'Device support' for more information.)

Panel Modularity

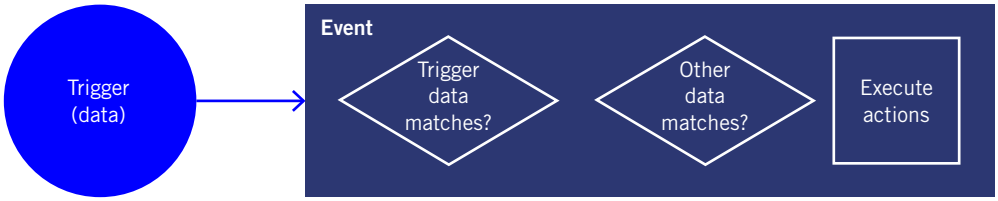
Similar to MDC Server, MDC Panel is also modular: the Panel base provides all the 'widgets' necessary to create a user interface. Buttons, text fields, lists, tabs, dialog boxes etc.

Some widgets are designed specifically for use with a Server module. For example, a Server crossbar module also includes a crossbar matrix widget for the user interface.

Automation and Control

Event Engine

The event engine simplifies the logic configuration for control and automation.
A sequence in the engine always follows this simple pattern:
Trigger [data] -> Condition(s) -> Action(s)



Event-based control

A trigger can be triggered by the MDC core, modules or an action. The data structure of triggers created by the core and modules are predefined, however, custom events can be defined with any data structure.

In the condition, data sent by the trigger can be checked. Other runtime data such as variables or switches can also be checked.

Lastly, you define which action is executed during a successful match. An action can (among other things) execute a command in a Server module, trigger a trigger or change the Panel status.

Panel

The MDC Panel is freely configurable and enables control of all aspects of MDC Server. The logic part remains on the Server – the Panel configuration describes the UI and its client-side functionality and defines the data binding to the Server data.

For the Server, a Panel, like any other infrastructure component, is a module instance. It is possible to address several panels in one instance as well as use different instances for individual panels (groups).

A server-side Panel cache ensures that all data is stored and updated, should the Panel become inaccessible or be switched off. For example, restarting the Panel automatically causes data to be read from the Server cache so the Panel is initialised correctly, without having to worry about configuration.

Panel

Stability and Performance

We believe that stability is the most important measure of an automation and control solution. Various safeguards in MDC ensure that failure of an infrastructure component affects the rest of the control as little as possible. A correctly configured MDC Server instance runs absolutely stably. The occurrence of an exception within a module (with the exception of the loss of that module functionality) has no effect on the remaining operation.

The event engine is multithreaded for performance reasons, and the framework ensures that thread lock or race conditions are avoided as much as possible. If threading problems arise due to device or connection problems or a faulty configuration, watchdogs ensure that the affected threads are stopped. It is therefore not necessary to consider event engine multithreading during event configuration.

Technology

All MDC framework components and modules are developed in C#.NET and can be run on standard Windows hardware. We recommend the following system configuration:

– Server: HP Proliant Server or equivalent, Microsoft® Windows™ Server 2008 R2 or later. Hyper-V virtualisation possible but not recommended.
– Panel: Desktop PC min. Windows® XP, recommended Windows® 7 +, X86 Tablets Windows™ 7 or later. Hyper-V virtualisation possible.
– Admin: Microsoft® Windows™ 7 or later. Hyper-V virtualisation possible.

The IP network is used exclusively for the communication between the MDC components Server, Admin and Panel. Server modules and Panel widgets can also communicate with the infrastructure via serial interface, GPIO, USB or any other interface.

MDC Server can either be run as a Windows service or run as a console application (for testing and configuration).

MDC Admin is a Windows desktop application.

MDC Panel and widgets are touchscreen/tablet-friendly. The Panel can be adapted for any resolution and executed in full-screen borderless/kiosk mode or as a normal desktop application and can be operated in a multimonitor configuration.

Device Support

MDC currently supports the following industry standard protocols:	**SNMP**	The SNMP module supports freely configurable OIDs and SNMP Set, Get and Traps.
	DMX	Freely configurable DMX channels per device, group support, fuse protection.
	Ember(+)	Any function of an Ember (+) device can be addressed. Support for Subscribe Messages. Stand-alone application for the discovery of Ember devices.
	ProBel	Support for the ProBel standard.
	SMTP	Standard SMTP mail module, support for authentication.
Utility Module:	**Alerts**	Exception-based alerting, such as via email.
	NetTrigger	With the NetTrigger module any external, string-based command can be filtered over TCP or UDP.
	Persistance	Server-side persistence module that can store any data.
Server module for proprietary protocols:	**OnAirXML**	Interface for David Software XML export.
	BlackBox KVM	Support for IP KVM from the company BlackBox.
	Brainstorm	Module for controlling Brainstorm VR systems.
	LED Display	Control for the most common LED matrix displays.
	Evertz Symphony	Support for the Evertz Symphony protocol.
	Evertz Xenon	Evertz Xenon crossbar support.
	EzTV	Control of EzTV boxes.
	GPIO	GPIO support for Siemens industrial control.
	Infratec Power	Control of Infratec Ethernet connector strips.

	Evertz Logo Inserter	Module for Evertz Logo Inserter.
	Nevion	Support for crossbars made by Nevion.
	Panasonic Camera	Control of Panasonic Pan & Tilt cameras.
	Panasonic Plasma	Control of Panasonic plasma displays.
	Panasonic Display	Control for various Panasonic LCD displays/PJ-Link.
	PowerSwitch	Control of PowerSwitch sockets.
	RDS Service	Control of standard RDS encoders.
	Scheduler	Simple scheduling service.
	SigInt	Control of signalling hardware of type SigInt.
	SmartView	Blackmagic SmartView analyser support.
	SoundWeb	Support for BSSaudio SoundWeb.
	Vidigo	Vidigo visual radio product support.
	And much more.	
Panel Widgets	**Broadcast Timer**	Visualisation of time periods, remaining time, time, etc.
	Faders	Customisable linear regulators.
	Matrix	Freely configurable matrix widget with multi-take for all supported audio and video crossbars.
	Profiles	Customisable client profiles, context-aware.

Any Questions?

www.danexis.com