SIEMENS

DMS8000 MP4.40
Generic DVR
Integration Guide

Building Technologies

Fire Safety & Security Product

Data and design subject to change without notice. / Supply subject to availability.

About this document

Purpose of this document

This manual is a guide to the development procedures for the integration of 3rd party video recorders in MM8000 systems.

This guide is en extension to the general DMS8000 documentation set that includes the Composer Technical Manual, the DMS8000 Connectivity Guides (mainly the Video and OPC guides, but also Network/Fire/Intrusion and Access), and the MM8000 Installation, Configuration, and Commissioning manual (ICC).

Target audience

This documentation is intended for the following users:

- Project Managers
- External System Engineers and Developers

Individuals performing the operations described in this manual are expected to be familiar with the video units to integrate and with the XML markup language.

Documentation resource information

The *DMS8000 Documentation Resource Information and Glossary Guide* assembles important information regarding documentation resources. This document contains the following:

- Comprehensive definitions of the target audiences for Siemens FS DMS documents
- Training program information including the Siemens intranet link
- A complete list of all available DMS8000 documents
- Instructions for how to obtain a document via the Siemens intranet using the Siemens Asset Portal
- A map of relevant documents for each target audience group
- Customer Support links & resources
- A glossary containing definitions of all terms and acronyms used in DMS8000 documentation

To access the *DMS8000 Documentation Resource Information and Glossary Guide* (document no. A6V10089056), go to the link and follow the document search instructions below:

http://assetportal.bt.siemens.com/portal/index.html

1. In the Search column on the left, set:
   - Segment: **04 Fire -3F**
   - Document Type: **All**
   - Image Type: **All**
   - Advanced search criterias: Select **Brochure No.** and enter the document number to search for (e.g. A6V10062415). Alternatively, select **Title** and enter the product name (e.g. MM8000).

2. Click **Search** to start.

3. In the resulting area on the right, select the document type (e.g. Contacts, Data Sheet, etc).

For more information such as Siemens news and announcements, visit the STEP Web portal at:

https://workspace.sbt.siemens.com/content/00001123/default.aspx

Liability disclaimer for damage or injuries

Before products are delivered, they are tested to ensure they function correctly when used properly. Siemens disclaims all liability for damage or injuries caused by the incorrect application of the instructions, or the disregard of danger advisories. This disclaimer applies in particular to personal injuries or damage caused by:

– Improper and/or incorrect use.

– Disregard of safety instructions in the documentation or on the product.

– Poor maintenance or a lack of maintenance.

We have checked the contents of this manual for agreement with the hardware and software described. Since deviations cannot be precluded entirely, we cannot guarantee full agreement. However, the data in this manual are reviewed regularly and any necessary corrections included in subsequent editions. Suggestions for improvement are welcome.

Modification index

Version	Date	Notes
A6V10331273_a_en	06.2011	Corresponds with DMS8000 MP4.40

1 Introduction

This is a technical guide to the integration of a DVR unit in MM8000. It provides the technical information that enables you to instruct MM8000 to handle a 3rd party video device for both video image display and status reporting.

2 Integrating a DVR

This chapter describes the MM8000 features that allows for integrating a 3rd party Digital Video Recorder (DVR).

Integrating a Digital Video Recorder in MM8000

The MM8000 systems MM8000 can support the connection to 3rd party DVRs in two different ways:

- **Light video integration: uncontrolled video viewer**

  This type of integration allows for starting a *video viewer* (a Windows application or a local HTML page) and then manually selecting the video sources that cannot be selected automatically depending on the MM8000 conditions. No status diagnostic can be provided.

  ➔ This solution does not require any special development and is fully described in the DMS8000 Connectivity Guides Video (doc.no. A6V10062457). Please refer to the *Light-integration device* section.

- **Generic DVR integration: controlled video viewer and OPC DA connection**

  This second type of integration, requiring the technical development described in this guide, permits to configure a customized DVR, including video and alarm I/O, and start the associated *video viewer* (a Windows application or a web page) with appropriate parameters when invoked. Depending on the viewer characteristics, the parameters can be used e.g. to select a video source or activate other options. The list of parameters includes MM8000 conditions as well as user-defined options that can be configured on site, thus supporting a large number of integration scenarios.

  ➔ This solution requires the development of an XML file that contains the customized model of the DVR (video and alarm inputs /outputs) and the viewer invocation parameters. Once the file is prepared and installed, the DVR units can be easily configured as described in the DMS8000 Connectivity Guides Video (doc.no. A6V10062457).

  The DVR can also provide status reporting and receive control commands over an *OPC DA connection*. MM8000 can act as OPC DA client to acquire diagnostic information and transmit control commands.

  ➔ The OPC connectivity is described in the DMS8000 Connectivity Guides OPC (doc.no. A6V10065253).

Generic DVR add-on distribution

This guide also discusses how to prepare a distribution package (MM8000 Add-on) of the files required for the Generic DVR integration. Refer to section 4 on p.19.

2.1 Understanding the Generic DVR ST

Composer environment

Composer is the common configuration tool environment that implements the DMS Engineering.

When an MM8000 station is installed at a site, it needs to be configured for the specific facility. The Composer software is developed for this purpose. It is a Windows application that provides the general services and a multi-area user interface as follows:

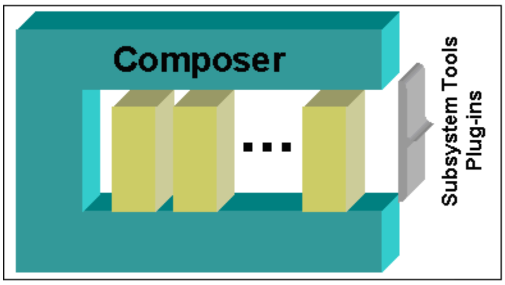
- The user interface and common functionality to configure a system
- The configuration environment that hosts Subsystem Tool plug-ins

Subsystem Tools

The specific functions of configuring the system components are implemented by software modules called Subsystem Tools (STs). Composer provides for an extension interface for adding and integrating plug-in modules.

The subsystem tools can be plugged into Composer to form an environment devoted to the configuration of the control units of a system. Depending on the number and kind of subsystem tools, this environment allows configuration of a single control unit up to a large security system including several control units of different disciplines.

The end user perceives a single application for the configuration activities; however the system is made up of different pieces of software: the Composer infrastructure and the subsystem tools, as illustrated by the following picture.



Generic DVR ST

Unlike standard STs that are specialized for supporting a single control unit type, the Generic DVR ST is a special subsystem tool (or meta-tool) that allows for supporting multiple, customized DVR unit models. The configured models can then be used for an easier configuration of DVR unit instances that fit one of the defined models.

The Generic DVR ST is therefore an open tool that supports multiple application models. This enables the engineer working with Composer to create DVR units (instances of a model) with the required set of parameters.

2.2 Generic DVR models

As discussed above, the Generic DVR ST can handle multiple DVR models. The corresponding files must exist in the folder:

<installation folder>\Composer\Configuration Data\Generic DVR Configuration Data\...

with < installation folder> being for example C:\Program Files\DMS8000 (in English editions of Windows).

DVR modelling types

Three types of DVR can be modelled depending on the type of video viewer. Namely:

- Windows application viewer: the video viewer is an installable Windows application.
- Web-based viewer type 1: the DVR provides a web server with a single URL and camera selection parameters.
- Web-based viewer type 2: the DVR provides a web server with multiple addresses corresponding with camera pages.

  ➔ Contact customer support for application examples:

DVR parameter files

Each valid DVR model comprises two files with the same name:

- The XML file described below (.XML)
- A standard icon file (.ICO, including sizes 16x16, 32x32 and 48x48 pixels)

All valid GENERIC DVR files that exist in the folder will be available as GENERIC DVR model in the Composer configuration environment (refer to the DMS8000 Connectivity Guides Video, doc.no. A6V10062457).

Tip: in defining a new model, start from en existing profile and modify it as necessary.

2.3 3rd Party DVR Integration Concept

The generic DVR integration is based on the following approach:

- The controlled activation (with parameters) of the 3rd party video viewer in the MM8000 environment. The viewer can be:
  - A Windows application (EXE file) providing the display functions. The technical requirements that this application must match are detailed in chapter 3 at p.16.
  - A web page (or pages) providing the display functions.

  The viewer characteristics must be defined in the *XML definition file* (see 2.4 below).
- The OPC DA data exchange of diagnostic reporting and control commands. The 3r party DVR operates as OPC DA server. Data points representing the diagnostic information must be defined in the *XML definition file* (see 2.4 below).

  The OPC connectivity to data points must be defined using the standard MM8000 OPC client configuration.

2.4 XML definition file

The XML file defining the DVR model contains the following information:

- The control information about the viewer and the related invocation parameters, which are necessary to select the appropriate video source depending on the condition in the management station. Three types of scenarios are foreseen:
  - The viewer is a Windows application that can handle in-line command parameters, e.g.:

    C:\ProgramFiles\VGMaker\VGViewer.exe -s1 -c12
  - The DVR provides a web server to access video images that is invoked using a general URL with sub-paths that can select a specific video source, e.g.:

    http://www.intranet1/vgmaker/vg.aspx#?page=viewer&camera=112
  - The DVR provides a web server to access video images that is invoked using a specific URL for each video source, e.g.:

    rtsp://www.intranet1/vgmaker/vg/cam112dome/video.gci
- The maximum number (0-256) of Alarm Inputs, Alarm Outputs, Video Inputs, and Video Outputs. Zero means that no inputs of that type are foreseen.
- The DVR datapoint structure to be inserted in Composer, including folders for the configurable inputs and outputs.

The XML file can include the following node structures (XML node classes):

DVR (mandatory, including viewer and limits) …	Node TYPE 1
VideoIn Folder ……………………………………….	Node TYPE 20
VideoIn (predefined video inputs) ……………….	Node TYPE 21
VideoOut Folder …………………………………….	Node TYPE 30
VideoOut (predefined video outputs) …………….	Node TYPE 31
AlarmInput Folder ………………………………….	Node TYPE 50
AlarmInput (predefined alarm inputs) …………….	Node TYPE 51
AlarmOutput Folder	Node TYPE 60
AlarmOutput (predefined alarm outputs) ………..	Node TYPE 61

2.4.1 DVR settings (XML node TYPE 1)

Model name

The XML file must define the name of the DVR used in Composer to select the corresponding model. The XML structure is the following:

```
<!– DRV description ->
  <property ID="CustomerText">
    <item VAL="DVR model XYZ" />
  </property>
  <property ID="TechText">
    <item VAL="DVR model for MM8000 developed on … by …" />
  </property>
```

According to the Composer requirements, both a *Customer Text* and a *Technical Text* must be specified.

2.4.1.1 Video viewer

Windows application or web page

The XML file must define whether the viewer is a Windows application or a web page. Optionally, a version can also be indicated.

The corresponding XML structure is the following (*Windows Application V1.0* in the example):

```
<property ID="ParametersList">
<!-- Mandatory field:  1 = Application - 2 = Web  -->
<Type Value="1"></Type>
<Version Value="1.0"></Version>
```

Application name

In case of Windows application, the command must include the application pathname, e.g.:

C:\Program Files\VGMaker\VGViewer.exe

The corresponding XML structure is the following (*editable in Composer with a help note thanks to the RW Screen Attribute and to the Note attribute*):

```
<!--  Mandatory fields for Application Integration  -->
<Application ScreenAttribute="RW" DefaultValue="C:\Program
Files\VGMaker\VGViewer.exe" Note="Please enter the
application pathname"> </Application>
```

Web page

In case of web server, note the *WebAddressType*: 1 means that a single web address is available, whereas 2 means that each video input requires a different web address (see section 2.4.3 at p.4). The XML structure for type 1 is the following (*visible in Composer with a help note thanks to the R Screen Attribute and to the Note attribute*):

```
<!--  Mandatory fields for Web Integration  -->
<WebAddressType Type="1" Note="1 - Main address with parame-
ter 2 - Custom Address for each Video Input" />
<WebAddress ScreenAttribute="R" Val-
ue="http://www.intranet1/vdmaker/vd.aspx#?page=viewer&camera=
%%VIDEOIN_NO%%)%% Note="Web Address depending on selected
video input." />
```

ScreenAttribute and Note XML attributes

In the example above, note the *ScreenAttribute* option that can be:

“H” = Hidden, not visible in the Composer page.

“R” = Read, visible in the Composer page.

“W” = Write, editable in the Composer page.

The *Note* attribute is used to provide a description of the field. In case the *ScreenAttribute* option “R” or “RW” is selected, the *Note* field is also available in the Composer configuration page.

2.4.1.2 Invocation Line Parameters

The invocation line can include parameters that appear in the command line or in the URL. The parameter list must be defined according to the viewer requirements.

Parameters can be defined as fixed or variable text. For instance, a parameter such as **-Cnn** that indicates the camera number (for example “–C12”) can include the fixed prefix **-C** and a variable part that defines the actual video input number (see VIDEOIN_NO below) according to the selection in the runtime software.

Types of variable parameters

In the invocation line, variable parameters can be inserted using the *%%variable%%* syntax. Three types of fields are foreseen:

- Custom Variables, defined by the user in the XML file, e.g.: **%%MYVAR%%**
- System Variables, automatically provided by the system based on runtime conditions, e.g.: **%%VIDEOIN_NO%%**
- System Conditions, or "if-clauses", determined by the system based on runtime conditions, e.g.: **%%IF_PTZ(…)%%**

Custom Variables

The *Custom Variables* are created for user-defined configuration in Composer, for example (*editable IP address with a default and a help note*):

```
<variable Name="IP" ScreenAttribute="RW" Description="IP Address" DefaultValue="192.168.1.1" MinValue="" MaxValue=""
Note=" Please enter the DVR IP address" />
```

Custom variables with *ScreenAttribute*=RW (Read/Write) can be edited in Composer starting from the *DefaultValue*, within the limits defined by *MinValue* and *MaxValue*, and using *Description* and *Note* information as user help.

System Variables

The list of *System Variables* includes:

- VIDEOIN_NO – video input number (e.g. "1" or "112")
- DATE_TIME – date and time (CTime structure: "%d/%m/%Y - %H:%M")

  ➔ See http://msdn.microsoft.com/en-us/library/fe06s4ak(v=VS.71).aspx.

  (e.g.: "DATE_TIME" → "16/03/2011 – 13:52")
- DATE_TIME[CustomFormatString] – customized date and time

  ➔ See http://msdn.microsoft.com/en-us/library/fe06s4ak(v=VS.71).aspx.

  (e.g.: "DATE_TIME[%m/%d/%Y - %H:%M]" → "03/16/2011 – 13:52")

  For all formats: in the event treatment, dates refer to the event's timestamp, whereas in the plant browser, the current date & time is used.
- PARENT_HND – parent handle window
- POS_X – screen position top left
- POS_Y – screen position top left
- SIZE_X – window size width
- SIZE_Y – window size height

⚠ System Variables names cannot be used for Custom Variables.

System Conditions

The list of valid *System Conditions* includes:

- IF_PB(…) – If Plant Browser is active, then …
- IF_ET(…) – If Event Treatment is active, then …
- IF_PTZ(…) – If camera supports PTZ commands and PTZ control is selected on the user interface toolbar, then …
- IF_N_PTZ(…) – If camera does not support PTZ commands or PTZ control is not selected on the user interface toolbar, then …
- IF_LIVE (…) – If the software is displaying live video, then …

– IF_PLAYBK(…) – If the software is displaying playback video, then …

For any variable, note that the “%%” string is used as separator to indicate both the beginning and the end of variables or conditions.

Here below is an XML example of parameter configuration including all types of variables. The viewer expects up to 3 parameters. The first parameter (/t=…) is user defined and can be different depending on the MM8000 condition (plant browser navigation or event treatment). The second parameter (/in=…) is the video input number associated with the current selection in MM8000. The third parameter (“/p”) may or may not be present, depending whether camera is motorised and can support PTZ control commands.

```
<CommandLine Value= " "
"/t=%%IF_PB(%%CUST_PARAM_SURVEILLANCE%%)%%%%IF_ET(%%CUST_PARA
M_EVENT%%)%% /in=%%VIDEOIN_NO%%%%IF_PTZ( /p)%%" />
```

I/O limits

The range of the I/O limit must be between 0 and 256. If =0, no instance of that type of objects will be used.

The corresponding XML structure can be the following (*in the example, three groups out of four are foreseen with different limits*):

```
<!-- Mandatory fields - System limits-->
<AlarmInputMax  Value="2" Note="Max 2 alarm inputs" >
</AlarmInputMax>
<AlarmOutputMax Value="1" Note="Max 1 alarm output">
</AlarmOutputMax>
<VideoInputMax  Value="4" Note="Max 4 video inputs">
</VideoInputMax>
<VideoOutputMax Value="0" Note="No video outputs">
</VideoOutputMax>
```

Tip: in Composer (set in Debug mode using a register command in **…\DMS8000\Utilities\Reg\ST_Debug_Yes.reg**), on the **Custom Settings** tab of the DVR, you can check the invocation string for the selected model.

The Windows event viewer can show the resulting string in the command line after the variables interpretation (refer to 3.2.2 at p.17).

2.4.2 Folders (XML node TYPE 20, 30, 50, 60)

Folders for inputs and outputs with max. value >0 must be defined in the XML so that they will be part of the Composer model applied whenever a DVR instance is inserted in the Composer configuration.

For example, the XML structure for the video input folder can be:

```
node ID="1000" TYPE="20">
<properties>
  <property ID="CustomerText">
    <item VAL="Video Inputs Folder" />
  </property>
  <property ID="TechText">
    <item VAL="Video-in Folder" />
  </property>
</properties>
```

The *TYPE* and *ID* attributes must correspond with the folder type:

- Video Input Folder: TYPE 20, ID 1000

- Video Output Folder: TYPE 30, ID 2000
- Alarm Input Folder: TYPE 50, ID 3000
- Alarm Output Folder: TYPE 60, ID 4000

According to the Composer requirements, both a *Customer Text* and a *Technical Text* must be specified.

2.4.3 I/O nodes (XML node TYPE 21, 31, 51, 61)

Individual inputs and outputs points may or may not be predefined in the XML because they can also be inserted in Composer as many times as required up to the defined limit.

For example, the XML structure for a video output can be:

```
node ID="2001" TYPE="31">
<properties>
  <property ID="CustomerText">
    <item VAL="Video Output #1" />
  </property>
  <property ID="TechText">
    <item VAL=" Video-out 1" />
  </property>
</properties>
```

The *TYPE* and *ID* attributes must correspond with the I/O type:

- Video Input Folder: TYPE 21, ID 1001-1256
- Video Output Folder: TYPE 31, ID 2001-2256
- Alarm Input Folder: TYPE 51, ID 3001-3256
- Alarm Output Folder: TYPE 61, ID 4001-4256

According to the Composer requirements, both a *Customer Text* and a *Technical Text* must be specified.

Individual video-in web address

When the model is set for using a custom address for each video input (web address type = 2), a web address is required for each video In. In this type of configuration, the *WEBAddress* is a property of each video input node.

The following XML property line can be added to each video input to predefine the web address:

```
<property ID=" VideoInputWEBAddress">
  <item VAL="rtsp://www.intr1/vgmaker/cam112dome/video.gci"/>
</property>
```

Users can enter, if not predefined, or modify the predefined address in Composer.

2.4.4 Examples of XML file

2.4.4.1 Windows application viewer

This example refers to a Windows application viewer that is started with a command line such as:

C:\Program Files\VGMaker\VGViewer.EXE -AP1 –AR1 –I192.168.9.112 –C12 –TViewer –W120,200,100,100 -P

In Composer, the Program file pathname as well as the following parameters can be configured:

- PTZ control enabled (-AP)
- Preset Position enabled (-AR)
- DVR IP address

In addition, the DVR support 4 video inputs, which are all declared and instantiated in the Composer project, and 2 optional alarm inputs that can be added manually in the dedicated folder.

```
<?xml version="1.0" encoding="iso-8859-1" ?>
<subsystems>
   <subsystem>
      <nodes>
         <node ID="1" TYPE="1">
            <properties>
               <property ID="CustomerText">
                  <item VAL="Generic Viewer" />
               </property>
               <!--  DRV description  -->
               <property ID="TechText">
                  <item VAL="DVR Viewer Application" />
               </property>
               <property ID="ParametersList">
                  <!--  Mandatory fields = 1 = Application - 2 = Web   -->
                  <Type Value="1" />
                  <Version Value="1" />
                  <Application ScreenAttribute="RW"
                  DefaultValue="C:\Program Files\VGMaker\VGViewer.EXE"
                  Note="Please insert here your application path" />
                  <CommandLine Value="-AP%%PTZ%% -AR%%PP%% -I%%IP%% C%%VIDEOIN_NO%%
                  -T%%APP_WINCTRL_TITLE%% -W%%POS_X%%,%%POS_Y%%,%%SIZE_X%%,%%SIZE_Y%%
                  %%IF_LIVE(%%IF_PTZ( -P)%%)%%"/>
                  <!-- Mandatory fields - System limits (Not visible in Composer) -->
                  <AlarmInputMax Value="2" Note="Max Alarm Input value" />
                  <AlarmOutputMax Value="0" Note="Max Alarm Output value" />
                  <VideoInputMax Value="4" Note="Max Video Input value" />
                  <VideoOutputMax Value="0" Note="Max Video Output value" />
                  <!-- Custom Variable definition - System limits  -->
                  <variable Name="PTZ" ScreenAttribute="RW" Description="PTZ control"
                  DefaultValue="1" MinValue="" MaxValue="" Note="Enable PTZ control" />
                  <variable Name="PP" ScreenAttribute="RW" Description="Preset Position"
                  DefaultValue="1" MinValue="" MaxValue="" Note="Enable preset control" />
                  <variable Name="IP" ScreenAttribute="RW" Description="DVR Address"
                  DefaultValue="192.168.1.101" MinValue="" MaxValue="" Note="IP address." />
                  <variable Name="APP_WINCTRL_TITLE" ScreenAttribute="H"
                  Description="Main Window application Title" DefaultValue="Viewer" MinValue="0"
                  MaxValue="0" Note="Window title used for detecting 3rd party main window" />
               </property>
            </properties>
            <node ID="1000" TYPE="20">
               <properties>
                  <property ID="CustomerText">
                     <item VAL="Folder" />
                  </property>
                  <property ID="TechText">
                     <item VAL="Video Inputs Folder" />
                  </property>
               </properties>
               <node ID="1001" TYPE="21">
                  <properties>
                     <property ID="CustomerText">
                        <item VAL="Input 1" />
```

See below an example of resulting string.

```
                     </property>
                     <property ID="TechText">
                        <item VAL="Video Inputs 1" />
                     </property>
                  </properties>
               </node>
               <node ID="1002" TYPE="21">
                  <properties>
                     <property ID="CustomerText">
                        <item VAL="Input 2" />
                     </property>
                     <property ID="TechText">
                        <item VAL="Video Inputs 2" />
                     </property>
                  </properties>
               </node>
               <node ID="1003" TYPE="21">
                  <properties>
                     <property ID="CustomerText">
                        <item VAL="Input 3" />
                     </property>
                     <property ID="TechText">
                        <item VAL="Video Inputs 3" />
                     </property>
                  </properties>
               </node>
               <node ID="1004" TYPE="21">
                  <properties>
                     <property ID="CustomerText">
                        <item VAL="Input 4" />
                     </property>
                     <property ID="TechText">
                        <item VAL="Video Inputs 4" />
                     </property>
                  </properties>
               </node>
            </node>
            <node ID="3000" TYPE="50">
               <properties>
                  <property ID="CustomerText">
                     <item VAL="Folder" />
                  </property>
                  <property ID="TechText">
                     <item VAL="Alarm Inputs Folder" />
                  </property>
               </properties>
            </node>
         </node>
      </nodes>
   </subsystem>
</subsystems>
```

Note that, assuming the following runtime conditions:

- Live image view
- Selected video in = 12
- PTZ camera and PTZ controls enabled on user interface (toolbar button)

the resulting invocation string is:

```
C:\Program Files\VGMaker\VGViewer.EXE -AP1 –AR1 –I192.168.9.112 –C12 –TViewer –W120,200,100,100 -P
```

2.4.4.2 Web-based viewer type 1

This example refers to a Web-based viewer that is started with a URL address including a common part and a different suffix for selecting various video sources. For example:

http://www.siemens.com/milansite/vipiteno/videosurveillance/cam#01.html

The suffix #01 must change to #02, #03 and so on to address other video sources.

The Composer project includes a predefined folder for the video inputs (up to 16) that have to be manually added according to the site configuration

```
<?xml version="1.0" encoding="iso-8859-1" ?>
<subsystems>
   <subsystem>
      <nodes>
         <node ID="1" TYPE="1">
            <properties>
               <property ID="CustomerText">
                  <item VAL="Web generic viewer" />
               </property>
               <!--  DRV description  -->
               <property ID="TechText">
                  <item VAL="DVR Web Based type 1" />
               </property>
               <property ID="ParametersList">
                  <!--  Mandatory fields = 1 = Application - 2 = Web   -->
                  <Type Value="2" />
                  <Version Value="1" />
                  <!-- Mandatory fields - System limits (Not visible in Composer) -->
                  <AlarmInputMax Value="0" Note="Max Alarm Input value" />
                  <AlarmOutputMax Value="0" Note="Max Alarm Output value" />
                  <VideoInputMax Value="16" Note="Max Video Input value" />
                  <VideoOutputMax Value="0" Note="Max Video Output value" />
                  <!-- Mandatory fields for WEB Type Integration -->
                  <WebAddressType Type="1" Note="1 - Main address with parameter
                                                 2 – Custom Address for each Video Input" />
                  <!--  URL definition -->
                  <WebAddress ScreenAttribute="R" Value="%%COMMON_ADDRESS%%%%VIDEOIN_NO%%.html
                  Note="Intranet Web Address defined using the system variable for input number" />
                  <!--  Custom Variable definition -->
                  <variable Name="COMMON_ADDRESS" ScreenAttribute="RW"
                  Description="Common intranet address" DefaultValue="
                  http://www.siemens.com/milansite/vipiteno/videosurveillance/cam#"
                  MinValue="" MaxValue="" Note="Please insert the common part of video URL." />
               </property>
            </properties>
            <node ID="1000" TYPE="20">
               <properties>
                  <property ID="CustomerText">
                     <item VAL="Folder" />
                  </property>
                  <property ID="TechText">
                     <item VAL="Video Inputs Folder" />
                  </property>
               </properties>
            </node>
         </node>
      </nodes>
   </subsystem>
</subsystems>
```

2.4.4.3 Web-based viewer type 2

This example refers to a Web-based viewer that is started with a URL line that is configured individually in Composer for each video input, up to 255 inputs.

The Composer project includes a predefined folder for the video inputs that have to be manually added according to the site configuration

```
<?xml version="1.0" encoding="iso-8859-1" ?>
<subsystems>
   <subsystem>
      <nodes>
         <node ID="1" TYPE="1">
            <properties>
               <property ID="CustomerText">
                  <item VAL="Web generic viewer" />
               </property>
               <!--  DRV description  -->
               <property ID="TechText">
                  <item VAL="DVR Web Based type 2" />
               </property>
               <property ID="ParametersList">
                  <!--  Mandatory fields = 1 = Application - 2 = Web   -->
                  <Type Value="2" />
                  <Version Value="1" />
                  <!-- Mandatory fields - System limits (Not visible in Composer) -->
                  <AlarmInputMax Value="0" Note="Max Alarm Input value" />
                  <AlarmOutputMax Value="0" Note="Max Alarm Output value" />
                  <VideoInputMax Value="255" Note="Max Video Input value" />
                  <VideoOutputMax Value="0" Note="Max Video Output value" />
                  <!-- Mandatory fields for WEB Type Integration -->
                  <WebAddressType Type="2" Note="1 - Main address with parameter
                                                 2 – Custom Address for each Video Input" />
               </property>
            </properties>
            <node ID="1000" TYPE="20">
               <properties>
                  <property ID="CustomerText">
                     <item VAL="Folder" />
                  </property>
                  <property ID="TechText">
                     <item VAL="Video Inputs Folder" />
                  </property>
               </properties>
            </node>
         </node>
      </nodes>
   </subsystem>
</subsystems>
```

3 3rd Party Application Requirements

This chapter presents the technical requirements that a 3rd party software application must match to be integrated in MM8000 as a video (or more general data) viewer.

3.1 Technical Requirements for Windows Application Viewer

Referring to the types of supported viewers described in 2.3 at p.6, these requirements concern the Windows application. This application must:

- Supplied as a valid executable Windows application (**EXE file**) that is invoked by MM8000 to start the viewer.
- Support **command-line parameters**. Applying this technique, MM8000 provides the viewer with the parameters defined in the XML definition files (see 2.4 at p. 6).
- Handle **one main window** according to Windows application design standards. That window will be used by MM8000 to control the viewer application.
- Avoid displaying any **splash screen** upon starting up. Such splash screens would be quite annoying to users.
- Support the start, stop and window control methods described below.

Windows start methods

MM8000 must be able to identify the viewer main window in the list provided by the *EnumThreadWindows* function. The viewer must support at least one of the following ways techniques (listed according to their priority):

- Defining a custom variable named “APP_WINCTRL_CLASS” and assign the class name of the main window as default value.
- Defining a custom variable named “APP_WINCTRL_TITLE” and assign the title of the main window as default value. Note that the caption text must be fixed.
- Ensure that the main window has the following style settings:
  - WS_CAPTION or WS_OVERLAPPED
  - WS_EX_WINDOWEDGE

MM8000 expects the process to activate within 3 seconds and the main window to start within 5 seconds from program invocation. Error logs are generated in case any timeout expires (refer to 3.2.2 at p.17).

Windows close methods

MM8000 must be able to close the viewer application. The viewer must support at least one of the following techniques:

- Closing upon receiving the standard WM_CLOSE message.
- Closing upon receiving the WM_ApplicationName_CLOSE message.
  With *ApplicationName* being the executable filename without the EXE extension.

After a 3 second wait, if the viewer application does not terminate, MM8000 stops the application (process) using the *TerminateProcess* windows function. An error log is generated in case the timeout expires (refer to 3.2.2 at p.17).

Windows control methods

MM8000 must be able to resize, move and close the 3rd party viewer that support the following Windows control messages:

- WM_SIZE
- WM_MOVE
- WM_CLOSE

Refer to 3.2.3 at p.18 for links to the Windows development library.

3.2 Reference tables

3.2.1 Timeouts

Description timeout	Value (sec)	Meaning
Timeout to create the process.	3	Timeout expired waiting for idle input for the new created process.
Timeout to search application’s main window.	5	The timeout expires when the main window is not found (for the Application=%s identified by the ProcessId=xx and the ThreadId=xx) within the maximum delay from activation. In this case, the external application cannot be controlled.
Timeout to close the application.	5	The timeout expires when the application does not terminate within the maximum delay from the closing command. In this case, the external application is forced to terminate.

When a timeout expires, a specific event log is added to Windows Event Viewer with event source = GenericDVRViewer.

3.2.2 Event Logs

When	Message	Parameters	Type
Process created	Process created with command line ss	ss: string (command parameters in command line).	Information

Main window not found	Main window not found for Application=ss ProcessId=xx ThreadId=xx after the timeout tt msec. The external application cannot be controlled.	ss = Application Name
		xx = Process & thread ID
		tt = Timeout
			Warning

Process closing (custom message)	Error detected sending message to the external application. Handle=hh	hh = windows ID	Warning
Process closing (standard message)	Error detected sending WM_CLOSE message to the external application. Handle=hh	hh = windows ID	Warning
Closing timeout expired	Closing timeout expired. The external application will be forced to terminate.		Warning

Application’s main window detected	The external application main window is: Hwnd=hh ProcessId=xx ThreadId=xx
	Detected by the Class cc
		hh = Windows ID
		xx = Process & thread ID
		cc = Windows Class name
			Information

Application’s main windows detected	The external application main window is: Hwnd=hh ProcessId=xx ThreadId=xx
	Detected by the Title tt
		hh = Windows ID
		xx = Process & thread ID
		cc = Windows Title
			Information

Application’s main windows detected	The external application main window is: Hwnd=hh ProcessId=xx ThreadId=xx
	Detected by Windows Styles (WS_CAPTION,WS_OVERLAPPED and WS_EX_WINDOWEDGE)
		hh = Windows ID
		xx = Process & thread ID
			Information

3.2.3 Windows control messages

Windows Message	**Description**
WM_SIZE	http://msdn.microsoft.com/en-s/library/ms632646(v=vs.85).aspx
WM_MOVE	http://msdn.microsoft.com/en-s/library/ms632631(v=vs.85).aspx
WM_CLOSE	http://msdn.microsoft.com/en-us/library/ms632617(v=VS.85).aspx

4 Creating an MM8000 Add-on distribution package

4.1 Add-on installation kit

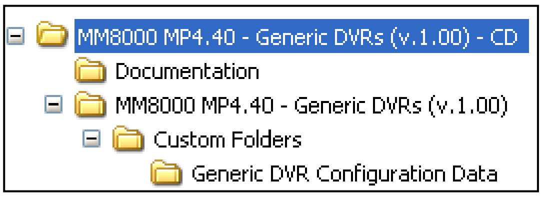
The MM8000 files for configuring a Generic DVR are distributed as an add-on kit to be installed after the standard MM8000 software setup. Starting from an add-on kit template, you can create the add-on installation kit with your own files and easily install your Generic DVR configuration package on each MM8000 system that requires it.

➔ Contact the DMS8000 Customer Support to get the add-on installation kit template for Generic DVR.

4.2 Generic DVR add-on creation

Add-on installation kit template

The installation kit template for Generic DVR includes the following folders (Fig 1):



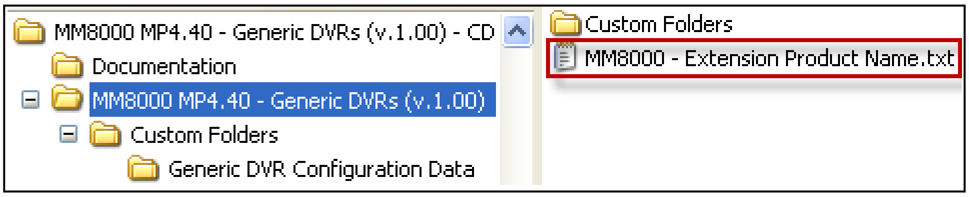
MM8000 MP4.40 - Generic DVRs (v.1.00) - CD
Documentation
MM8000 MP4.40 - Generic DVRs (v.1.00)
Custom Folders
Generic DVR Configuration Data

Fig 1 Add-on installation kit

Starting from the installation kit template, the Generic DVR add-on preparation requires the following steps:

1. Define Add-on name and version

   Edit the text file **MM8000 - Extension Product Name.txt** (Fig 2) in the **MM8000 MP4.40 - Generic DVRs (v.1.00)** folder



MM8000 MP4.40 - Generic DVRs (v.1.00) - CD
Documentation
MM8000 MP4.40 - Generic DVRs (v.1.00)
Custom Folders
Generic DVR Configuration Data

Custom Folders
MM8000 - Extension Product Name.txt

Fig 2 Text file with installation kit name and version

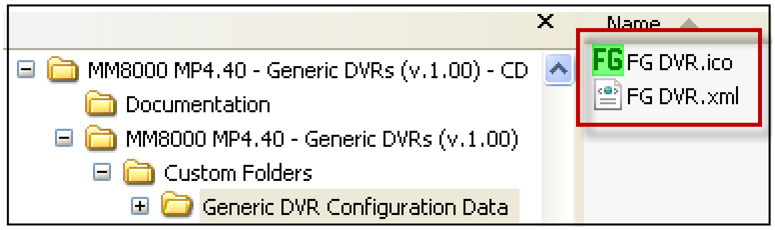
The text file includes the following lines:

```
[GENERAL]

ADDON ID=1.00
ADDON NAME=FG DVR
VERSION MAJOR=4
VERSION MINOR=4
ADDON COMPATIBILITY VER = 3
```

You can customize the **ADDON ID** (version) and the **ADDON NAME**. Be careful not to modify any other line.

2. Add your DVR(S) parameter files (XML and ICO pairs, refer to 2.2 on p.5) in the **Generic DVR Configuration Data** folder


Name
MM8000 MP4.40 - Generic DVRs (v.1.00) - CD
Documentation
MM8000 MP4.40 - Generic DVRs (v.1.00)
Custom Folders
Generic DVR Configuration Data
FG DVR.ico
FG DVR.xml

The **Generic DVR Configuration Data** folder should contain one more file pairs corresponding with one more DVR models that will appear in the Composer menu for Generic DVRs.

➔ *The add-on installation kit is ready.*

4.3 Generic DVR add-on Installation

A specific application, the Add-on manager, is provided by MM8000 to install and update the add-on modules.

The following describes the installation procedure.

1. Start Add-on Manager.

From the Windows Start menu, select the following:

Start → DMS8000 → Tools → Add-on Manager

The Add-on Manager window appears.

2. Click the **Browse** … button, locate the installation files of the add-on module and select the text file (**Extension Product Name.txt**)

3. If your MM8000 installation includes a customized MM8000 internal account, then deselect the checkbox **Use default MM8000 User account** and specify the customized username and password.

4. Click **Install** (or **Update** if a previous version is detected).