



Security Center SDK Release Notes 5.3 GA

Click [here](#) for the most recent version of this guide.

Copyright notice

© 2015 Genetec Inc. All rights reserved.

Genetec Inc. distributes this document with software that includes an end-user license agreement and is furnished under license and may be used only in accordance with the terms of the license agreement. The contents of this document are protected under copyright law.

The contents of this guide are furnished for informational use only and are subject to change without notice. Genetec Inc. assumes no responsibility or liability for any errors or inaccuracies that may appear in the informational content contained in this guide.

This publication may not be copied, modified, or reproduced in any form or for any purpose, nor can any derivative works be created therefrom without Genetec Inc.'s prior written consent.

Genetec Inc. reserves the right to revise and improve its products as it sees fit. This document describes the state of a product at the time of document's last revision, and may not reflect the product at all times in the future.

In no event shall Genetec Inc. be liable to any person or entity with respect to any loss or damage that is incidental to or consequential upon the instructions found in this document or the computer software and hardware products described herein. The use of this document is subject to the disclaimer of liability found in the end-user license agreement.

"Genetec", "Omnicast", "Synergis", "Synergis Master Controller", "AutoVu", "Federation", "Stratocast", the Genetec stylized "G", and the Omnicast, Synergis, AutoVu, and Stratocast logos are trademarks of Genetec Inc., either registered or pending registration in several jurisdictions.

"Security Center", "Security Center Mobile", "Plan Manager", "Sipelia", and the Security Center logo are trademarks of Genetec Inc.

Other trade names used in this document may be trademarks or registered trademarks of the manufacturers or vendors of the respective products.

All specifications are subject to change without notice.

Document information

Document title: Security Center SDK Release Notes 5.3 GA

Document number: EN.501.001-V5.3.B(2)

Document update date: April 30, 2015

You can send your comments, corrections, and suggestions about this guide to documentation@genetec.com.

Contents

Security Center SDK Release Notes 5.3 GA

What's new in Security Center 5.3 SDK	2
Dynamic alarms	2
RTSP Media Router role	5
SDK video overlay	7
User and user group configuration	10
Video quality configuration	12
Motion detection configuration	14
Active Directory configuration	16
Visual tracking configuration	18
Media player	19
Synchronous playback	20
Scheduled task configuration	20
Event-to-action configuration	22
Partitions	24
Resolved issues in Security Center 5.3 SDK	26
Known issues in Security Center 5.3 SDK	27
Known limitations in Security Center 5.3 SDK	28

Security Center SDK Release Notes 5.3 GA

Security Center is the unified security platform that seamlessly blends Genetec's IP security and safety systems within a single innovative solution. The systems unified under Security Center include Genetec's Omnicast IP video surveillance system, Synergis IP access control system, and AutoVu IP license plate recognition (LPR) system.

This document describes the new features that were added in the Security Center 5.3 SDK. It also describes the known limitations and provides code samples when appropriate.

NOTE Plan Manager provides its own SDK documentation.

What's new in Security Center 5.3 SDK

The following capabilities are now available in the Security Center 5.3 SDK:

- "Dynamic alarms" on page 2
- "RTSP Media Router role" on page 5
- "SDK video overlay" on page 7
- "User and user group configuration" on page 10
- "Video quality configuration" on page 12
- "Motion detection configuration" on page 14
- "Active Directory configuration" on page 16
- "Visual tracking configuration" on page 18
- "Media player" on page 19
- "Synchronous playback" on page 20
- "Scheduled task configuration" on page 20
- "Event-to-action configuration" on page 22
- "Partitions" on page 24

IMPORTANT This is the last version of the Security Center SDK implemented on .NET Framework 4.0. Subsequent versions will target .NET Framework 4.5. Therefore, this is the last version of Security Center that supports Windows XP and Windows Server 2003.

Dynamic alarms

You can use the SDK to trigger alarms in Security Center with additional (dynamic) content without modifying the properties of the source alarm entity. It is also possible to retrieve the information transmitted by a dynamic alarm using the query *AlarmActivityQuery*. Note that this feature is available only through the SDK.

Using the *DynamicAlarmContent* class, you can specify the following:

- **AttachedEntities.** Entities that are appended (not added) to the already existing set of entities attached to the alarm entity.
- **ForwardedRecipients.** Recipients to whom the alarm is forwarded. When the alarm is triggered, it is forwarded to those recipients only when they are logged in. The forwarded recipients are also appended to the list of recipients already attached to the alarm entity.
- **Priority.** Overrides the priority of the source alarm entity.
- **Context.** Information that helps the user to understand the purpose of the alarm.
- **URLs.** List of URLs that the recipients can access when receiving the alarm.

Limitations

This SDK feature has the following limitations:

- Dynamic alarms behave like forwarded alarms. They are not persisted.
 - Users who are not logged in when a dynamic alarm is triggered will not receive the alarm.
 - Users who close and reopen their Security Desk will not see past dynamic alarms anymore.
 - Dynamic alarms will appear in the alarm reports only if the dynamic content is applied to an existing or a newly created alarm entity.

Sample code

The following two new methods allow you to trigger a dynamic alarm by GUID or by alarm entity:

```

/// <summary>
/// Creates an active instance of the specified alarm.
/// </summary>
/// <remarks>This alarm will wait for the given acknowledgement condition to be cleared before
being acknowledged</remarks>
/// <param name="alarm"> Alarm identifier</param>
/// <param name="sourceEntityGuid"> Source entity's GUID</param>
/// <param name="dynamicAlarmContent"> Dynamic alarm content to attach to the alarm instance</
param>
/// <exception cref="ArgumentOutOfRangeException"> If alarm GUID is empty</exception>
/// <returns> Alarm instance ID</returns>
public int TriggerAlarm( Guid alarm, Guid sourceEntityGuid, DynamicAlarmContent
dynamicAlarmContent)

/// <summary>
/// Creates an active instance of the specified alarm
/// </summary>
/// <remarks> This alarm will wait for the given acknowledgement condition to be cleared before
being acknowledged</remarks>
/// <exception cref="ArgumentNullException"> If alarm is null</exception>
/// <param name="alarm"> Alarm entity</param>
/// <param name="sourceEntityGuid"> Source entity's GUID</param>
/// <param name="dynamicAlarmContent"> Dynamic alarm content to attach to the alarm instance</
param>
/// <returns> Alarm instance ID</returns>
public int TriggerAlarm( Alarm alarm, Guid sourceEntityGuid, DynamicAlarmContent
dynamicAlarmContent)

```

The following method shows a typical usage of *DynamicAlarmContent*:

```

/// <summary>

```

```

/// Creates and triggers a dynamic alarm using dynamic alarm content.
/// </summary>
private void CreateDynamicAlarm()
{
    //create or retrieve the alarm entity that we want to trigger with dynamic content
    Alarm alarm = m_engine.CreateEntity("My Dynamic Alarm", EntityType.Alarm) as Alarm;
    alarm.ReactivationThreshold = TimeSpan.Zero;
    Guid createdAlarm = alarm.Guid;

    //get the User to notify
    User user= m_engine.GetEntity(new Guid("373b3403-a9c0-4bb1-ad2b-43327d8e19ca")) as User;

    //get the entity
    Camera camera = m_engine.GetEntity(
        new Guid("00000001-0000-babe-0000-2ea7b2eb70e4")) as Camera;

    //create the dynamic alarm content
    DynamicAlarmContent dac = new DynamicAlarmContent("My Context");

    //add a URL to help the recipients with the alarm triggered
    dac.Urls.Add("www.genetec.com");

    //set the alarm priority, overrides the parent's alarm priority
    dac.Priority = 5;

    //attach the entities
    dac.AttachedEntities.Add(camera.Guid);

    //attach the recipients
    dac.ForwardedRecipients.Add(user.Guid);

    //this will trigger the alarm with its dynamic content
    int alarmResult = m_engine.AlarmManager.TriggerAlarm(alarm, user.Guid, dac);

    if (alarmResult == -1)
        PostMessage("Alarm was not triggered");
}

/// <summary>
/// Creates the alarm activity query.
/// </summary>
/// <remarks>
/// Using this query, we do not have access to the recipients of the DynamicAlarmContent
/// </remarks>
private void CreateAlarmActivityQuery()
{
    var query = m_engine.ReportManager.CreateReportQuery(ReportType.AlarmActivity) as
        AlarmActivityQuery;
    query.Alarms.Add(new Guid("373b3303-a9c0-4bb1-ad2b-43327d8e19ca"));
}

```

```

//cast the result to an AlarmActivityQueryCompletedEventArgs
//we do not have access to the recipients of the DynamicAlarmContent triggered
AlarmActivityQueryCompletedEventArgs result = query.Query() as
    AlarmActivityQueryCompletedEventArgs;

if((result != null) && (result.Success))
{
    foreach (DataRow dr in result.Data.Rows)
    {
        int instanceId = (int)dr[0];
        Guid alarmGuid = (Guid)dr[1];
        Guid triggerEvent = (Guid)dr[2];
        string dynamicContext = (string)dr[16];

        //to retrieve the list of URLs
        foreach (DataRow row in result.DynamicUlrsData.Rows)
        {
            int urlInstanceId = (int)row[0];
            if (instanceId == urlInstanceId)
            {
                string url = (string)row[1];
            }
        }

        //to retrieve the list of attached entities
        foreach (DataRow row in result.DynamicAttachedEntitiesData.Rows)
        {
            int urlInstanceId = (int)row[0];
            if (instanceId == urlInstanceId)
            {
                Guid entityGuid = (Guid)row[1];
                Entity entity = m_engine.GetEntity(entityGuid);
            }
        }
    }
}

```

RTSP Media Router role

You can use the new RTSP Media Router role to access video streams through Security Center. Because the video streams are available through Security Center, you can take advantage of the routing and redirection capabilities, coupled with the flexibility of raw video received directly from the camera. In addition, you can display the video using any standard viewer that supports the camera's specific encoding. You could then use such a viewer to display video in a web page for example.

To use this feature:

- You need a valid license that includes one or more RTSP connections. Refer to the *Number of RTSP Streams* option in the Security Center license.
- You need to know the Globally Unique Identifier (GUID) of the camera to formulate an RTSP request. You can get the camera GUID through the Security Center SDK. However, for this to be available, you need a valid SDK license and a valid certificate.
- You must create and configure a *RTSP Media Router* role in Security Center.

NOTE The Security Center SDK is not required to send RTSP requests to the RTSP Media Router role, however it is needed to get the camera GUIDs to include in the requests.

Limitations

This feature has the following limitations:

- The RTSP Media Router role only supports RFC 2326.
- You can request either live or playback video but, non-video streams such as audio, PTZ commands, overlays and metadata are not supported.
- The username and password are associated with the RTSP Media Router role and not with the cameras. Anyone who has the username and password of the role can have access to any camera in the system regardless to their privileges.

URL formatting

URLs are formulated as follows:

```
rtsp://<IP>:<port>/<camera guid>/<stream> [? <attributes>]
```

Where you need to specify the following parameters:

- **<IP>**. The IP address of the RTSP Media Router role.
- **<port>**. The port number of the RTSP Media Router role. The port number can be set in Config Tool when initially creating the role.
- **<camera guid>**. The GUID of the requested camera. You can get this value through the *Camera.Entity* property available in the SDK.
- **<stream>**. The name of the stream to request. It can be *live*, *playback*, *highres*, *lowres*, or *remote* and it reflects the stream usages configured in Config Tool for the camera. Note that for playback, the requested time must be included in the body of the RTSP web request.

EXAMPLE RTSP requests typically look like the following examples:

- `rtsp://localhost:654/00000001-0000-babe-0000-1adc2042a2b4/live`
- `rtsp://10.2.111.203:654/00000001-0000-babe-0000-00408c730d81/playback`

Sample code

To send RTSP requests, you need to create and configure a RTSP Media Router role in Config Tool.

The following code sample shows how to format a RTSP request URL. See the SDK help and samples for information on querying entities.

```
private string CreateRtspUrl(Guid cameraId)
{
    //use one of the following 5 preconfigured streams: live, playback, highres,
    //lowres, remote
    string streamName = "live";

    //create the RTSP URL
    UriBuilder url = new UriBuilder();
    url.Host = "localhost";
    url.Port = 654;
    url.Path = cameraId + "/" + streamName;
    url.Scheme = "rtsp";
    return url.ToString();
}

private string CreateRtspUrl(string cameraName)
{
    var query = m_engine.ReportManager.CreateReportQuery(ReportType.EntityConfiguration)
        as EntityConfigurationQuery;
    query.Name = cameraName;
    query.EntityTypes.Add(EntityType.Camera);
    QueryCompletedEventArgs result = query.Query();
    var cameraId = (Guid) result.Data.Rows[0]["Guid"];
    return CreateRtspUrl(cameraId);
}

private string CreateRtspUrl(int cameraLogicalId)
{
    Camera camera = m_engine.GetEntity(EntityType.Camera, cameraLogicalId) as Camera;
    return CreateRtspUrl(camera.Guid);
}
```

SDK video overlay

You can use the Security Center SDK and the Web SDK to add textual and graphical overlays on displayed video. You can create overlays that are stored by the Archiver to be displayed on both the live and playback video streams, or to be exported with the video. The SDK supports .jpg and .bmp images, polygons, circles, lines, rectangles and text in any installed typeface or formatting.

For instance, you can add an image or a logo with transparency as a watermark over a video image or draw shapes on the video image as part of a video analytics integration. An option in the Security Desk allows the users to enable or disable the overlays.

The following visual elements are available:

- **Text.** Font, size, position, text color, background color, style, transparency.
- **Font.** All font types are supported. The default font is Arial.
- **Polygon.** Vertex positions, line color, fill color, line width, and transparency.
- **Circle.** Center position, radius, line color, fill color, line width, transparency.
- **Rectangle.** Vertex positions, line color, fill color, line width, and transparency.
- **Line.** Vertex positions, line color, line width, transparency.
- **Camera GUID.** Camera identifier on which overlays are defined.
- **Overlay layers.** Layers on top of other layers.

NOTE All positions are relative to allow resizing.

Limitations

This SDK feature has the following limitations:

- Overlays are not displayed in real time. There is a delay between the time an overlay is created and the time users can see it on the live video stream.
- When an overlay is displayed on a moving object, it can introduce jitter. The workaround is to increase the jitter buffer delay in Security Center.
- Overlays are not displayed on Mobile Clients.
- Overlays are not displayed on the Web Clients.
- Overlays are not displayed on the Genetec Video Player.
- Overlays cannot be defined with ghost cameras.
- Overlays are not available if the recording is done on an edge camera.

Sample code

The overlay consists of the following classes: *OverlayFactory*, *Overlay* and *Layer*.

```
/// <summary>
/// Creates the overlay on a video stream.
/// </summary>
private void CreateOverlay()
{
    OverlayFactory.Initialize(m_engine);

    //get the camera on which you want to draw the overlay
    Camera camera = (Camera)m_engine.GetEntity(
```

```

        new Guid("00000001-0000-babe-0000-00408c730d81"));

//create an overlay using a unique name, or
//retrieve a previously created overlay using its unique name
Overlay overlay = OverlayFactory.Get(camera, "overlay1");

//initialize the drawing surface height and the drawing surface width
overlay.Initialize(800,800);
Brush brush = new SolidColorBrush(Color.FromRgb(255,0,0)); //Red

//create the layer if it does not exist already
//retrieve the layer if it was created previously
//use both the name and the GUID to identify and retrieve the last created layer
Layer layer = overlay.CreateLayer(
    new Guid("1E9B49F5-644C-4F1D-962C-6DF8449FFBF9"), "layer");

//clear the layer just in case it previously contained drawings
layer.Clear();

//set a duration of 3 minutes
layer.Duration = TimeSpan.FromMinutes(3);
layer.DrawEllipse(brush,new Pen(brush, 15), new Point(250,150), 45, 45);

//update should be the last call
//leave a time in between to allow overlay transfer
layer.Update();

//the next line will dispose of the layer and will not show it on the video anymore
layer.Dispose();

//disposing of the overlay does not clear the drawing.
overlay.Dispose();
}

/// <summary>
/// Creates the overlay text.
/// </summary>
private void CreateOverlayText()
{
    OverlayFactory.Initialize(m_engine);

    //get the camera on which you want to draw the overlay
    Camera camera = (Camera)m_engine.GetEntity(
        new Guid("00000001-0000-babe-0000-00408c730d81"));

    //create an overlay using a unique name
    //or retrieve a previously created overlay using its unique name
    Overlay overlay = OverlayFactory.Get(camera, "overlay1");

```

```
//initialize the drawing surface height and the drawing surface width
overlay.Initialize(800, 800);

Brush brush = new SolidColorBrush(Color.FromRgb(255, 0, 0));

Layer layer = overlay.CreateLayer(
    new Guid("{525DBE6E-5579-4B36-AC88-761A646A5410}"), "Layer3");
layer.Clear();

Typeface myfont = new Typeface("Calibri");
FormattedText text = new FormattedText(DateTime.Now.Hour.ToString("00h"),
    CultureInfo.InvariantCulture, FlowDirection.LeftToRight, myfont, 40,
    new SolidColorBrush(Colors.LimeGreen));

layer.Duration = new TimeSpan(0, 0, 1, 0); //20 minutes
layer.Update(); leave a time in between to allow overlay transfer

//for this sample, dispose is called immediately after
//generally you should leave some time between update and dispose to allow overlay transfer
overlay.Dispose();
}
```

Using a *Layer*, it is also possible to call the following methods:

- DrawEllipse
- DrawImage
- DrawLine
- DrawRectangle
- DrawText

User and user group configuration

It is now possible to modify certain additional user and user group settings such as privileges. Methods have been added to set the following user and user group entity properties:

- ArchiveViewingLimitation
- IsArchiveViewingLimitationInherited
- SecurityLevel
- IsSecurityLevelInherited
- Privileges (including custom privileges)

Limitations

This SDK feature has the following limitations:

- You cannot add or remove privileges. You can only modify existing privileges.

- Some privileges are used to group other privileges. They can be used to set children privileges, but when reading the privilege configuration of an entity, such *parent* privileges always return **Undefined**. It differs from the Config Tool, which has built-in logic to adapt the display depending on the values of child privileges.

Sample code

The following code sample shows how to use this SDK feature.

```
/// <summary>
/// Changes the privilege of a user.
/// </summary>
private void ChangePrivilegeOfUser()
{
    User user = m_engine.GetEntity(
        new Guid("f69c0fe3-8e18-4cad-bae3-9df383456f16")) as User;

    //create a transaction to modify the user's privileges
    m_engine.TransactionManager.CreateTransaction();
    user.SetPrivilegeState(SdkPrivilege.RunSecurityDesk, PrivilegeAccess.Granted,
        PrivilegeChangeBehavior.ApplyToChildrenPrivileges);
    m_engine.TransactionManager.CommitTransaction(true);
}

/// <summary>
/// Changes the user's security level.
/// </summary>
private void ChangeUserSecurityLevel()
{
    //find the user for whom you want to modify the security level
    User user = m_engine.GetEntity(
        new Guid("f69c0fe3-8e18-4cad-bae3-9df383456f16")) as User;
    byte securitylevel = user.SecurityLevel;

    //create a transaction if there is more than one security level being changed at once
    m_engine.TransactionManager.CreateTransaction();
    user.SecurityLevel = 230;

    //for the newly set SecurityLevel to be effective,
    //you should not inherit them from the parent
    user.IsSecurityLevelInherited = false;
    m_engine.TransactionManager.CommitTransaction(true);
}

/// <summary>
/// Changes the user archive viewing limitation.
/// </summary>
private void ChangeUserArchiveViewingLimitation()
{

```

```

User user = m_engine.GetEntity(
    new Guid("f69c0fe3-8e18-4cad-bae3-9df383456f16")) as User;
TimeSpan tsAfter = new TimeSpan(10, 0, 0, 0);
user.ArchiveViewingLimitation = tsAfter;
user.IsArchiveViewingLimitationInherited = false;
}

```

Video quality configuration

You can now set some of the video quality and color settings associated with a camera. Some of the properties that you can now set through the SDK are:

- Resolution
- Constant bit rate
- Key frame interval
- Recording frame rate

Limitations

This SDK feature has the following limitations:

- You can only change the basic video quality settings of a camera. These settings are the most common among all the manufacturers. You cannot generally set manufacturer specific settings through the SDK.
- No validation is performed on the values associated with specific configuration settings for video quality. It is your responsibility to provide valid values through the SDK. For available options, refer to the Config Tool.

Sample code

The following code sample shows how to use this SDK feature.

Take note of the following:

- Video attributes (brightness settings) can be accessed from the *Camera* entity.
- Stream specific configuration and capabilities can be accessed through the *VideoStream* entity.
- The stream usages associated with a camera are accessed through the *Camera.StreamUsages* property.

```

/// <summary>
/// Sets the camera color settings.
/// </summary>
public void SetCameraColorSettings()
{
    //make sure that all the needed entities are in the entity cache before calling GetEntity

```

```

//call the GetEntity method to retrieve the camera
Camera camera = m_engine.GetEntity(
    new Guid("00000001-0000-babe-0000-00408c730d81")) as Camera;

//set the brightness, contrast, saturation and hue
int brightness=5;
int contrast = 10;
int saturation = 15;
int hue = 20;

//create video attributes with the correct color settings
VideoAttributes videoattributes = new VideoAttributes(Schedule.AlwaysScheduleGuid,
    brightness, contrast, saturation, hue);

//create a transaction to commit the changes
m_engine.TransactionManager.CreateTransaction();

//update the camera's color attributes
camera.UpdateVideoAttributesSchedule(Schedule.AlwaysScheduleGuid, videoattributes);

//commit the transaction
m_engine.TransactionManager.CommitTransaction();
}

/// <summary>
/// Sets the camera video quality.
/// </summary>
public void SetCameraStreamVideoQuality()
{
    //make sure that all the needed entities are in the entity cache before calling GetEntity
    //call the GetEntity method to retrieve the camera
    Camera camera = m_engine.GetEntity(
        new Guid("00000001-0000-babe-0000-00408c730d81")) as Camera;

    ReadOnlyCollection<Guid> streams = camera.Streams;

    foreach(Guid guid in streams)
    {
        //get the video streams
        VideoStream stream = m_engine.GetEntity(guid) as VideoStream;

        if (stream != null)
        {
            string name = stream.Name;

            if (name.Equals("MPEG-4 - 1"))
            {
                //this will set the recording frame rate of the stream

```

```
stream.SetRecordingFrameInterval(Schedule.AlwaysScheduleGuid,  
    RecordingFrameInterval.RecordAllFrames);  
  
//important to modify these settings in a transaction  
m_engine.TransactionManager.CreateTransaction();  
  
//set the key frame interval of the stream  
stream.SetKeyFrameInterval(Schedule.AlwaysScheduleGuid,10);  
  
//set the bit rate (the constant bit rate has to be enabled)  
stream.SetBitRate(Schedule.AlwaysScheduleGuid,18);  
  
//find all the available capabilities of the stream  
VideoCompressionCapabilities capabilities =  
    stream.VideoCompressionCapabilities;  
  
//retrieve the supported resolutions available  
List<StreamSupportedResolution> streamRes = capabilities.SupportedResolutions;  
  
if(streamRes.Count>0)  
{  
    //Set the stream resolution to the first available in the drop down menu  
    stream.SetCompressionResolution(Schedule.AlwaysScheduleGuid, streamRes[0]);  
}  
m_engine.TransactionManager.CommitTransaction(true);  
}  
}  
}
```

Motion detection configuration

You can now configure the following camera motion detection properties through the SDK:

- Motion detection state (enabled, disabled).
- Motion detection processing (Archiver, unit).
- Sensitivity (in Archiver processing only).
- Consecutive frame hits.
- Motion on and off thresholds.
- Motion zones (add, remove).
- Motion detection configuration schedule (add, remove).
- Define motion zones using the same tools available in Config Tool (pen, rectangle, eraser, fill, clear, invert).

Limitations

This SDK feature has the following limitations:

- Auto-calibrating the sensitivity is not possible when the motion detection is performed by the Archiver.
- The learning mode option is not supported.
- It is not possible to test a zone, to test all zones and to view all motion.

Sample code

The following code sample shows how to use this SDK feature.

```
/// <summary>
/// Changes the camera motion detection.
/// </summary>
public void ChangeCameraMotionDetection()
{
    //retrieve the camera for which you want to modify its motion detection settings
    Camera camera = m_engine.GetEntity(
        new Guid("00000001-0000-babe-0000-00408c730d81")) as Camera;

    //this will return a different MotionDetectionConfiguration for every schedule.
    ReadOnlyCollection<IMotionDetectionConfiguration> motionDetectionConfigurations =
        camera.MotionDetectionConfigurations;

    foreach(IMotionDetectionConfiguration configuration in motionDetectionConfigurations)
    {
        //modify the sensitivity
        configuration.Sensitivity = 23;

        //change the detection to be done on the unit
        configuration.MotionDetectionType = MotionDetectionType.Unit;
        configuration.ConsecutiveFrameHits = 3;

        //get all the motion zones of the motion detection.
        ReadOnlyCollection<IMotionDetectionZoneConfiguration> allzones = configuration.Zones;

        foreach (IMotionDetectionZoneConfiguration motion in allzones)
        {
            //change the zone configuration
            motion.MotionOnThreshold = 10;
            motion.MotionOffThreshold = 20;
        }
        //it is important to call this method for the changes to take place
        camera.UpdateMotionDetectionConfiguration(configuration);
    }
}
```

Active Directory configuration

You can now configure some options of the Active Directory role. You can connect to an Active Directory and retrieve the list of container objects and organizational units. You can add a group to the synchronized groups and for these groups, you can also configure the following properties:

- Synchronize as user group
- Create user on first logon
- Synchronize as cardholders
- Import credentials

Limitations

Other configuration settings of the Active Directory role are not supported through the SDK.

Sample code

The following code sample shows how to use this SDK feature.

```
/// <summary>
/// Adds or modifies an imported groups to the Active Directory.
/// </summary>
public void AddImportedGroups()
{
    int logicalID = 109;

    //make sure that this entity is loaded in the entity cache using an
    //EntityConfigurationQuery. Retrieve the Active Directory role
    ActiveDirectoryRole adRole = m_engine.GetEntity(EntityType.Role, logicalID)
        as ActiveDirectoryRole;

    if (adRole != null)
    {
        //get all the groups
        Collection<ImportedADGroupInfo> imported = adRole.ImportedGroups;

        foreach (ImportedADGroupInfo imp in imported)
        {
            //modify the properties of the already existing groups
            if (imp.GroupAccountName == "GroupName")
            {
                //modify the group here will only modify the copy
                //you need to set them to the ImportedGroups for the changes to take effect
                imp.SynchronizeUserGroup = true;
                imp.CreateUserOnFirstLogon = false;
                imp.SynchronizeCardholders = true;
                imp.SynchronizeCredentials = true;
            }
        }
    }
}
```

```

    }
}

//add a group to the Active Directory
ImportedADGroupInfo adgroupInfo = new ImportedADGroupInfo(new Guid(
    "{a6d70acd-4f89-4315-b276-b38ac82f0713}"), "SDK", true, true, false, false);

//add a group info to a copy of the collection
imported.Add(adgroupInfo);
m_engine.TransactionManager.CreateTransaction();

//modify the internal collection using the set
adRole.ImportedGroups = imported;

//commit the transaction
m_engine.TransactionManager.CommitTransaction(true);
}
}

/// <summary>
/// Connects and fetches a group from an Active Directory. Can be used to add imported groups.
/// </summary>
public void ConnectToActiveDirectory()
{
    ActiveDirectoryClient adClient = new ActiveDirectoryClient();
    adClient.ConnectionPath = "devtesting2.com";

    //should be set to false to use custom UserName and Password
    adClient.UseApplicationCredentials = false;
    adClient.UserName = "sdka";
    adClient.Password = "a";

    //the next boolean will return true if the connectionPath,
    //Username and password are set correctly
    bool verifyConnection = adClient.Connect();
    ActiveDirectoryContainerInfo info = adClient.GetDomainInfo();

    //get the container objects from the Active Directory
    ReadOnlyCollection<ActiveDirectoryContainerInfo> containerInfos =
        adClient.GetContainersInContainer(info.Guid);

    //this represents all the organizational unit available in the Active Directory.
    ReadOnlyCollection<ActiveDirectoryOrganizationalUnitInfo> unitinfos =
        adClient.GetOrganizationalUnitsInContainer(info.Guid);

    //you can retrieve the group info by name
    ActiveDirectoryGroupInfo groupinfoByName = adClient.GetGroupInfo("sdk2");

    //then you can retrieve the GUID of the group
    Guid guid = groupinfoByName.Guid;

```

```
//you can also retrieve the group info by GUID
ActiveDirectoryGroupInfo groupinfoByGuid = adClient.GetGroupInfo(guid);
}
```

Visual tracking configuration

You can configure the shapes used for visual tracking. You can:

- Add tracking zones with different shapes, rectangles and ellipses.
- Add an entity or a list of entities to a tracking zone.
- Clear entities from a tracking zone.
- Clear a specific tracking zone.
- Clear all tracking zones.
- Get entities associated with a tracking zone.
- Remove a single entity from a tracking zone.

For tracking zones, you can define the following attributes:

- **Shape.** Rectangle or ellipse.
- **Size.** Width and height.
- **Position.** X, Y and rotation.
- **Color.** Background color, border color, border thickness and opacity.

Limitations

As per known limitation 240792, *SetVisualTrackingConfiguration* is not supported for federated cameras. Configuration of those cameras can be read, but not set.

Sample code

The following code sample shows how to use this SDK feature.

```
/// <summary>
/// Clears the visual tracking zones and entities.
/// </summary>
public void ClearVisualTracking()
{
    Camera camera = m_engine.GetEntity(
        new Guid("00000001-0000-babe-0000-00408c730d81")) as Camera;

    VisualTrackingConfiguration config = camera.GetVisualTrackingConfiguration();
    config.ClearZones();
    camera.SetVisualTrackingConfiguration(config);
}
```

```

/// <summary>
/// Clears the entities visual tracking.
/// </summary>
private void ClearEntitiesVisualTracking()
{
    Camera camera = m_engine.GetEntity(
        new Guid("00000001-0000-babe-0000-00408c730d81")) as Camera;
    VisualTrackingConfiguration config = camera.GetVisualTrackingConfiguration();

    foreach (VisualTrackingZone zone in config.Zones)
    {
        zone.ClearEntities();
    }
    camera.SetVisualTrackingConfiguration(config);
}

/// <summary>
/// Creates the visual tracking zone.
/// </summary>
public void CreateVisualTrackingZone()
{
    Camera camera = m_engine.GetEntity(
        new Guid("00000001-0000-babe-0000-00408c730d81")) as Camera;
    VisualTrackingConfiguration config = camera.GetVisualTrackingConfiguration();

    VisualTrackingZone zone = new VisualTrackingZone();
    zone.RelativeWidth = 50;
    zone.RelativeHeight = 50;
    zone.RelativeXPosition = 50;
    zone.RelativeYPosition = 50;
    zone.StrokeThickness = 2;
    zone.ShapeType = VisualTrackingZoneShape.Rectangle;
    zone.Opacity = 40;
    zone.AddEntity(new Guid("00000001-0000-babe-0000-72d8b3ac33ef"));
    zone.AddEntity(new Guid("00000001-0000-babe-0000-3d922a97c8f3"));

    config.AddZone(zone);
    camera.SetVisualTrackingConfiguration(config);
}

```

Media player

The media player can be used to play video from Security Center inside your WPF application. Using the media player, users can select a recorded video for playback, perform fast forward and rewind at different speeds.

The media player provides the following new capabilities:

- It supports the g64x video file format for playback.

- It supports hardware acceleration to render video (QuickSync and Nvidia only).
- It can display overlays (MetadataStream).

The new *EncodedVideoSourceFilter* class gives access to the raw, undecoded frames. This class does not decode the video stream and does not render the video feed on screen.

NOTE In Security Center 5.3, when playing live video, the *MediaPlayer* component raises the *FrameRendered* event for each frame rendered by the control. In Security Center 5.2, it used to raise the *FrameRendered* event not more frequently than every 400 ms.

Limitations

This new SDK feature requires a valid license to use it with the hardware acceleration enabled.

Synchronous playback

The Media SDK can be used to play back video from multiple cameras simultaneously in your .NET application. You can attach multiple cameras to the *MediaPlayerSynchronizer* and force them to synchronize their video streams so that each tile will display video that was recorded at the same time. Changing the state of one media player affects the state of the other media players associated with the same *MediaPlayerSynchronizer*.

The following change was made to the synchronous playback feature:

- There is an option to enable the command synchronization mode in which the commands are sent to all media players at once. In this mode the media players will receive a *play* or *pause* command at the same time and the video streams are allowed to drift from one another.

Limitations

There are no limitations with this SDK feature.

Scheduled task configuration

A scheduled task is a special type of entity used to automate repetitive system functions such as triggering a buzzer or synchronizing an Active Directory. You can now create scheduled tasks through the SDK and do the following actions:

- Create a scheduled task with a recurrence.
- Attach an action to a scheduled task. The action can be one of the following:
 - *TriggerAlarmAction*
 - *AddBookmarkAction*
 - *ProtectLiveVideoAction*
 - *StartRecordingAction*

- StopProtectLiveVideoAction
- StopRecordingAction
- ArmIntrusionAreaAction
- ArmZoneAction
- DisarmIntrusionAreaAction
- DisarmZoneAction
- DisplayOnAnalogMonitorAction
- ExecuteMacroAction
- SetThreatLevelAction
- TriggerOutputAction

Limitations

This SDK feature has the following limitations:

- Not all actions are supported. See the SDK documentation for the list of supported actions.
- You cannot activate or deactivate a scheduled task through the SDK.

Sample code

The following code sample shows how to use this SDK feature.

You need to do the following:

- 1 Create a *ScheduledTask* entity.
- 2 Use one of the *ActionManager.BuildAction* methods to create an action.
- 3 Assign the action to the new *ScheduledTask* entity.

```
/// <summary>
/// Creates a the schedule task.
/// </summary>
public void CreateScheduleTask()
{
    //create an action using the BuildAction method.
    //make sure that this action is compatible with the Scheduled Task
    TriggerAlarmAction action = m_engine.ActionManager.BuildAction(
        ActionType.TriggerAlarm, new Guid("373b3303-a9c0-4bb1-ad2b-43327d8e19ca"),
        Guid.Empty) as TriggerAlarmAction;

    //create a ScheduledTask entity
    ScheduledTask scheduledTask = m_engine.CreateEntity("ScheduleTask123",
        EntityType.ScheduledTask) as ScheduledTask;

    //create a daily recurrence for the scheduled task
    DailySchedule daily = new DailySchedule();
```

```

daily.Hour = 3;
daily.Minute = 59;
daily.Second = 39;

//associate the recurrence with the scheduled task
scheduledTask.SetSchedule(daily);

//associate the action with the scheduled task
scheduledTask.SetAction(action);

//In Config Tool, your scheduled task should be created and configured properly.
}

```

Event-to-action configuration

Event-to-action is a feature that allows administrators to configure actions that are triggered when events occur in the system. You can now do the following operations through the SDK:

- Create and add an *EventToAction* to an entity. The action can be linked to a schedule.
- Remove an *EventToAction* from an existing entity.
- Change the action of an existing *EventToAction* that belongs to an entity.
- Remove all the *EventToAction* from an entity.

Limitations

This SDK feature has the following limitation:

- Not all events and actions are supported for all entity types. See the SDK documentation for the list of supported events and actions.

Sample code

The following code sample shows how to use this SDK feature.

You need to do the following:

- 1 Get an entity from the engine.
- 2 Use one of the *ActionManager.BuildAction* methods to create and configure an action.
- 3 Assign the action to the entity by adding it to *Entity.EventToActions*.
- 4 Get the set of events supported by an entity using the *Entity.SupportedEvents* property.

```

/// <summary>
/// Creates and adds an event-to-action.
/// </summary>
public void CreateEventToAction()
{
    //retrieve the entity that has the event-to-actions

```

```

User admin = m_engine.GetEntity(new Guid("00000000-0000-0000-0000-000000000003")) as User;

//use this entity to retrieve the event-to-actions associated with "All Entities" from
//Config Tool
SystemConfiguration systemconfiguration = m_engine.GetEntity(
    new Guid("00000000-0000-0000-0000-000000000007")) as SystemConfiguration;

Schedule weekly = m_engine.GetEntity(
    new Guid("54274602-0175-4eef-9c27-9904d3642104")) as Schedule;

//create the action using BuildAction
TriggerAlarmAction action = m_engine.ActionManager.BuildAction(ActionType.TriggerAlarm,
    new Guid("373b3303-a9c0-4bb1-ad2b-43327d8e19ca"), weekly.Guid) as TriggerAlarmAction;

//retrieve the EventToActionCollection from the entity
EventToActionCollection eventToActionCollection = admin.EventToActions;

//this can be used to retrieve the EventToActions from "All Entities".
//"All Entities" in Config Tool represent the SystemConfiguration entity.
EventToActionCollection systemConfigurationCollection =
    systemconfiguration.EventToActions;

//add the event and the action to the collection
EventToAction eventToAction = eventToActionCollection.Add(EventType.UserLoggedIn, action);
}

/// <summary>
/// Removes the event-to-action.
/// </summary>
public void RemoveEventToAction()
{
    //retrieve the entity that has the event-to-action
    User admin = m_engine.GetEntity(new Guid("00000000-0000-0000-0000-000000000003")) as User;

    EventToActionCollection eventToActionCollection = admin.EventToActions;
    EventToAction toDelete = null;

    foreach (EventToAction ev in eventToActionCollection)
    {
        if(ev.Action is TriggerAlarmAction)
        {
            toDelete = ev;
        }
    }
    if(toDelete!=null)
    {
        //The Remove method cannot be called while iterating through the collection.
        eventToActionCollection.Remove(toDelete);
    }
}

```

}

Partitions

You can use partitions to group entities, such as cameras, into separate security and organizational sets. Only authorized users of a partition can access the entities assigned to it.

You can now perform the following operations on partitions through the SDK:

- Create or delete a partition.
- Add or remove a child partition to/from a parent partition.
- Add or remove a user to/from the list of authorized users.
- Add or remove a user group to/from the list of authorized users.
- Add or remove an entity to/from a partition.

CreationPartitions contains partitions to which entities are automatically assigned when they are created. You can now assign an unlimited number of partitions to it. In the previous version of the SDK, you could only assign a maximum of 3 partitions to *CreationPartitions*.

Limitations

There are no limitations with this SDK feature.

Sample code

The following code sample shows how to use this SDK feature.

NOTE Note the following:

- *partition.AcceptedUsers* and *partition.AcceptedUsersCollection* are methods now deprecated in favor of *UserAccesses*.
- *UserAccesses* returns all the users and user groups that have access to a partition.
- You can get the parent partition of a partition by calling the method *GetPartitions* on the partition entity.
- You can get all the children partitions of a partition using the *Members* collection.
- *AddUserAccess* allows a user or a user group to access a partition so that they can see the entities assigned to the partition.
- *RemoveUserAccess* removes a user or a user group from having access to a partition.
- Adding and removing user access to a partition also adds and removes the access to child partitions.
- Non-administrative users cannot see the partition and its assigned entities unless they are given access with *AddUserAccess*.

```

/// <summary>
/// Creates a partition and adds a member to it.
/// </summary>
public void CreateAndAddMemberToPartition()
{
    //retrieve the GUID of the user
    Guid userGuid = new Guid("1471e417-bcd0-46ea-9dd9-c0d0238e07a1");

    //to insert a child partition in a parent partition, it has to be done through a
    //transaction
    m_engine.TransactionManager.CreateTransaction();

    Partition partitionParent = (Partition)m_engine.CreateEntity("Partition Parent",
        EntityType.Partition);

    Partition partitionChild = (Partition) m_engine.CreateEntity("Partition Child",
        EntityType.Partition);

    //the child gets inserted into the partition of the parent.
    partitionChild.InsertIntoPartition(partitionParent.Guid);
    m_engine.TransactionManager.CommitTransaction(true);

    //this adds a user to the parent partition
    partitionParent.AddUserAccess(userGuid);

    //the user should be added to the UserAccesses if it is not already there
    //(in case of an orphan user only)
    ReadOnlyCollection<Guid> userAccesses = partitionParent.UserAccesses;

    //remove a usergroup from the partition.
    partitionParent.RemoveUserAccess(new Guid("3dfc4eee-08bb-44a9-a12b-c5243de0a465"));

    //get a cardholder
    Cardholder cardholder = m_engine.GetEntity(
        new Guid("2e5eaf21-1f13-4524-ac67-b37bf05c1b2b")) as Cardholder;

    //add it as a member of the partion
    partitionParent.AddMember(cardholder);
}

```

Resolved issues in Security Center 5.3 SDK

The following software issues were resolved in Security Center 5.3 SDK.

Issue	Description
118495	You cannot create or modify custom fields using the web SDK.
124028	Querying the system using CredentialConfigurationQuery to find unassigned visitors does not scale well if you have a system with large number of credentials.
172262	SDK MediaPlayer: OpenFile doesn't throw an exception with nonexistent files.
172266	SDK MediaPlayer: PlayerStateChanged event does not raise Stopping and Stopped notification when calling the Stop method.
172272	SDK Media Player: Calling PlayArchive (from MediaPlayer, MediaPlayerSynchronizer, or on federated cameras) will return a NoVideoSequenceAvailable and no frames are rendered and no video is streamed.
172279	SDK MediaPlayer: When using a mediaplayersynchronizer with registered Media players, if you call Stop on one of the Media players, the other Media player will also sync and stop, but the mediaplayer synchronizer state will not change.
172290	SDK MediaPlayer: When in Paused state, calling Seek or Rewind on a Media player will return the state NoVideoSequenceAvailable.
185083	If you create a custom incident report through the SDK, you cannot query for the custom incident fields in the Incidents report.
206955	When you create a zone through the SDK, you cannot select a role to manage the entity.
210680	Invalid values in certain SDK enumerations. IMPORTANT Because the fixes introduce breaking changes in enumeration values, applications that use the following enumerations need to be adjusted for the new values and recompiled: • Genetec.Sdk.Diagnostics.Logging.Core.LogSeverity • Genetec.Sdk.NetworkTransports • Genetec.Sdk.StreamConnectionType • Genetec.Sdk.IntrusionUnitCapabilities
211707	When doing a VideoEventQuery or any query from a derived class, the event identifiers in the resulting dataset are using the internal EventType enumeration instead of the SDK EventType enumeration. IMPORTANT Because the enumeration values returned by the query are now of the correct type, applications that use these values need to be adjusted and recompiled.
214955	SDK: There is no way to determine if a card format is a native format or a custom card format. The name credential is empty for both.

Known issues in Security Center 5.3 SDK

The Security Center 5.3 SDK includes the following known issues.

Issue	Description
146304	SDK: The entity cache does not update the cardholder's groups collection after a modification. When adding a CardholderGroup to a Cardholder, the CardholderGroup in the entity cache does not recognize that it has one more child. Workaround: Manually add the Cardholder to the CardholderGroup, or run an EntityConfigurationQuery to see the update. <pre>//Add a parent (CardholderGroup) to a child (Cardholder) cardholder.Groups.Add(cardholderGroup.Guid); //The CardholderGroup, which is now the parent of the cardholder, does //not recognize yet that it has one more member CardholderCollection collection = cardholderGroup.Members; //Workaround //Manually add the cardholder to the parent. //This is only useful to update the entity cache, so that there is no //inconsistency between parent and child //If this line is omitted, the user can still do an //EntityConfigurationQuery to update the entity cache. cardholderGroupRemoved.Members.Add(cardholder.Guid);</pre>
148015	<i>PtzIdleDelay</i> and <i>PtzLockDelay</i> on cameras will accept invalid values which can cause unexpected behaviors.
187732	It is possible to create a visitor with an activation date that exceeds the expiration date.
263582	Some Security Center federated cameras are listed as cameras that support overlays. However, Security Center federated cameras do not support overlays.

Known limitations in Security Center 5.3 SDK

The Security Center 5.3 SDK includes the following known limitations.

Issue	Description
222856	SDK: If a door lock or door unlock notification is raised from the SDK, the event will not be displayed in the Security Desk Monitoring task.
240792	<i>SetVisualTrackingConfiguration</i> is not supported for federated cameras. Configuration of those cameras can be read, but not set.

Where to find product documentation

You can find our product documentation in the following locations:

- **Installation package.** The documentation is available in the *Documentation* folder of the installation package. Some of the documents also have a direct download link to the latest version of the document.
- **Genetec Technical Assistance Portal (GTAP).** The latest version of the documentation is available from the GTAP [Documents](#) page. Note, you'll need a username and password to log on to GTAP.
- **Help.** Security Center client and web-based applications include help, which explain how the product works and provide instructions on how to use the product features. Patroller and the Sharp Portal also include context-sensitive help for each screen. To access the help, click Help, press F1, or tap the ? (question mark) in the different client applications.

Technical support

Genetec Technical Assistance Center (GTAC) is committed to providing its worldwide clientele with the best technical support services available. As a Genetec customer, you have access to the Genetec Technical Assistance Portal (GTAP), where you can find information and search for answers to your product questions.

- **Genetec Technical Assistance Portal (GTAP).** GTAP is a support website that provides in-depth support information, such as FAQs, knowledge base articles, user guides, supported device lists, training videos, product tools, and much more.

Prior to contacting GTAC or opening a support case, it is important to look at this website for potential fixes, workarounds, or known issues. You can log in to GTAP or sign up at <https://gtap.genetec.com>.

- **Genetec Technical Assistance Center (GTAC).** If you cannot find your answers on GTAP, you can open a support case online at <https://gtap.genetec.com>. For GTAC's contact information in your region see the Contact page at <https://gtap.genetec.com>.

NOTE Before contacting GTAC, please have your System ID (available from the About button in your client application) and your SMA contract number (if applicable) ready.

- **Licensing.**
 - For license activations or resets, please contact GTAC at <https://gtap.genetec.com>.
 - For issues with license content or part numbers, or concerns about an order, please contact Genetec Customer Service at customerservice@genetec.com, or call 1-866-684-8006 (option #3).
 - If you require a demo license or have questions regarding pricing, please contact Genetec Sales at sales@genetec.com, or call 1-866-684-8006 (option #2).

Additional resources

If you require additional resources other than the Genetec Technical Assistance Center, the following is available to you:

- **GTAP Forum.** The Forum is an easy to use message board that allows clients and Genetec staff to communicate with each other and discuss a variety of topics, ranging from technical questions to technology tips. You can log in or sign up at <https://gtapforum.genetec.com>.
- **Technical training.** In a professional classroom environment or from the convenience of your own office, our qualified trainers can guide you through system design, installation, operation, and troubleshooting. Technical training services are offered for all products and for customers with a varied level of technical experience, and can be customized to meet your specific needs and objectives. For more information, go to <http://www.genetec.com/Services>.