Simple API to access basic SourceAFIS functionality.
## Classes

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fingerprint</td>
<td>Collection of fingerprint-related information.</td>
</tr>
<tr>
<td>Person</td>
<td>Collection of <code>Fingerprint</code> belonging to one person.</td>
</tr>
<tr>
<td>AfisEngine</td>
<td>Methods and settings of SourceAFIS fingerprint matching engine.</td>
</tr>
</tbody>
</table>
# Enumerations

<table>
<thead>
<tr>
<th>Enumeration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Finger</td>
<td>Finger position on hand.</td>
</tr>
</tbody>
</table>

Send [feedback](mailto:) on this topic to the SourceAFIS maintainer.
Methods and settings of SourceAFIS fingerprint matching engine.

**Namespace:**  [SourceAFIS.Simple](#)

**Assembly:**  SourceAFIS (in SourceAFIS.dll)
Syntax

Visual Basic (Declaration)
Public Class AfisEngine

C#
public class AfisEngine

Visual C++
public ref class AfisEngine

JavaScript
SourceAFIS.Simple.AfisEngine = function();
Type.createClass(
    'SourceAFIS.Simple.AfisEngine');
Remarks

This class is an entry point to core SourceAFIS functionality. After setting relevant properties (notably **Threshold**), application can call one of the three main methods (**Extract(Person)**, **Verify(Person, Person)**, **Identify(Person, IEnumerable<Person>[])**) to perform template extraction and fingerprint matching.

AfisEngine objects are thread-safe, i.e. synchronized. AfisEngine is a lightweight object, but application is encouraged to keep only one global AfisEngine instance anyway. Every AfisEngine method utilizes multiple cores automatically. Applications that wish to execute several methods of AfisEngine in parallel should create multiple AfisEngine objects, perhaps one per thread.
Inheritance Hierarchy

System...Object
SourceAFIS.Simple...AfisEngine
See Also

AfisEngine Members
SourceAFIS.Simple Namespace

Send feedback on this topic to the SourceAFIS maintainer.
The AfisEngine type exposes the following members.
## Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AfisEngine</td>
<td>Create new SourceAFIS engine.</td>
</tr>
</tbody>
</table>
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identify</td>
<td>Compares one Person against a set of other Persons and returns best matches.</td>
</tr>
<tr>
<td>Verify</td>
<td>Compute similarity score between two Persons.</td>
</tr>
<tr>
<td>Extract</td>
<td>Extract fingerprint template(s) to be used during matching.</td>
</tr>
</tbody>
</table>
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Dpi</strong></td>
<td>Get/set DPI for processed images.</td>
</tr>
<tr>
<td><strong>MinMatches</strong></td>
<td>Get/set minimum number of fingerprints that must match in order for a whole person to match.</td>
</tr>
<tr>
<td><strong>Threshold</strong></td>
<td>Get/set similarity score threshold.</td>
</tr>
</tbody>
</table>
See Also

AfisEngine Class
SourceAFIS.Simple Namespace

Send feedback on this topic to the SourceAFIS maintainer.
Create new SourceAFIS engine.

**Namespace:**  [SourceAFIS.Simple](#)

**Assembly:**  SourceAFIS (in SourceAFIS.dll)
Syntax

**Visual Basic (Declaration)**

Public Sub New

**C#**

public AfisEngine()

**Visual C++**

public:
AfisEngine()

**JavaScript**

SourceAFIS.Simple.AfisEngine = function();
See Also

AfisEngine Class
SourceAFIS.Simple Namespace

Send feedback on this topic to the SourceAFIS maintainer.
The `AfisEngine` type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identify</td>
<td>Compares one Person against a set of other Persons and returns best matches.</td>
</tr>
<tr>
<td>Verify</td>
<td>Compute similarity score between two Persons.</td>
</tr>
<tr>
<td>Extract</td>
<td>Extract fingerprint template(s) to be used during matching.</td>
</tr>
</tbody>
</table>
See Also

AfisEngine Class
SourceAFIS.Simple Namespace

Send feedback on this topic to the SourceAFIS maintainer.
Extract fingerprint template(s) to be used during matching.

**Namespace:**  [SourceAFIS.Simple](#)  
**Assembly:**  SourceAFIS (in SourceAFIS.dll)
Syntax

Visual Basic (Declaration)

Public Sub Extract ( _
    person As Person _) _
)

C#

public void Extract(
    Person person
)

Visual C++

public:
void Extract(
    Person^ person
)

JavaScript

function extract(person);

Parameters

person
    Type: SourceAFIS.Simple....Person
    Person object to use for template extraction.
Remarks

Extract(Person) method takes Image from every Fingerprint in person and constructs fingerprint template that it stores in Template property of the respective Fingerprint. This step must be performed before the Person is used in Verify(Person, Person) or Identify(Person, IEnumerable<Of <<'(Person)>>)) method, because matching is done on fingerprint templates, not on fingerprint images.

Fingerprint image can be discarded after extraction, but it is recommended to keep it in case the Template needs to be regenerated due to SourceAFIS upgrade or other reason.
See Also

AfisEngine Class
SourceAFIS.Simple Namespace
AfisEngine...Dpi

Send feedback on this topic to the SourceAFIS maintainer.
SourceAFIS Fingerprint Recognition System
AfisEngine..::.Identify Method
AfisEngine Class  See Also  Send Feedback

Compares one Person against a set of other Persons and returns best matches.

Namespace:  SourceAFIS.Simple
Assembly:  SourceAFIS (in SourceAFIS.dll)
**Syntax**

Visual Basic (Declaration)

```vbnet
Public Function Identify (_
    probe As Person, _
    candidates As IEnumerable(Of Person) _
) As IEnumerable(Of Person)
```

C#

```csharp
public IEnumerable<Person> Identify(
    Person probe,
    IEnumerable<Person> candidates
)
```

Visual C++

```cpp
public: IEnumerable<Person>^ Identify(
    Person^ probe,
    IEnumerable<Person>^ candidates
)
```

JavaScript

```javascript
function identify(probe, candidates);
```

**Parameters**

probe
Type: `SourceAFIS.Simple...:..Person`  
Person to look up in the collection.

candidates
Type: `System.Collections.Generic...:..IEnumerable<Of <(Person)>>`  
Collection of persons that will be searched.
Return Value

All matching Person objects in the collection or an empty collection if there is no match. Results are sorted by score in descending order. If you need only one best match, call FirstOrDefault(Of <<'(TSource)>>)(IEnumerable(Of <<'(TSource)>>)) method on the returned collection.
Remarks

Compares probe Person to all candidate Persons and returns the most similar candidates. Calling Identify(Person, IEnumerable<Of <<'(Person)>>)) is conceptually identical to calling Verify(Person, Person) in a loop except that Identify(Person, IEnumerable<Of <<'(Person)>>)) is significantly faster than loop of Verify(Person, Person) calls. If there is no candidate with score at or above Threshold, Identify(Person, IEnumerable<Of <<'(Person)>>)) returns empty collection.

Most applications need only the best match, which can be obtained by calling FirstOrDefault<Of <<'(TSource)>>)(IEnumerable<Of <<'(TSource)>>)) method on the returned collection. Matching score for every returned Person can be obtained by calling Verify(Person, Person) on probe Person and the matching Person.

Persons passed to this method must have valid Template for every Fingerprint, i.e. they must have passed through Extract(Person) method.
See Also

AfisEngine Class
SourceAFIS.Simple Namespace
AfisEngine::Threshold
AfisEngine::MinMatches
AfisEngine::Verify(Person, Person)

Send feedback on this topic to the SourceAFIS maintainer.
Compute similarity score between two **Persons**.

**Namespace:**  **SourceAFIS.Simple**  
**Assembly:**  **SourceAFIS** (in SourceAFIS.dll)
Syntax

Visual Basic (Declaration)

Public Function Verify ( _
    probe As Person, _
    candidate As Person _
) As Single

C#

public float Verify(
    Person probe,
    Person candidate
)

Visual C++

public:
float Verify(
    Person^ probe,
    Person^ candidate
)

JavaScript

function verify(probe, candidate);

Parameters

probe
    Type: SourceAFIS.Simple...: Person
    First of the two persons to compare.

candidate
    Type: SourceAFIS.Simple...: Person
    Second of the two persons to compare.
Return Value

Similarity score indicating similarity between the two persons or 0 if there is no match.
Remarks

Verify(Person, Person) method compares two Persons, Fingerprint by Fingerprint, and returns floating-point similarity score that indicates degree of similarity between the two Persons. If this score falls below Threshold, Verify(Person, Person) method returns zero.

Persons passed to this method must have valid Template for every Fingerprint, i.e. they must have passed through Extract(Person) method.
See Also

AfisEngine Class
SourceAFIS.Simple Namespace
AfisEngine......Threshold
AfisEngine......MinMatches
AfisEngine......Identify(Person, IEnumerable<Of <<'(Person)>>))

Send feedback on this topic to the SourceAFIS maintainer.
The **AfisEngine** type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dpi</td>
<td>Get/set DPI for processed images.</td>
</tr>
<tr>
<td>MinMatches</td>
<td>Get/set minimum number of fingerprints that must match in order for a whole person to match.</td>
</tr>
<tr>
<td>Threshold</td>
<td>Get/set similarity score threshold.</td>
</tr>
</tbody>
</table>
See Also

AfisEngine Class
SourceAFIS.Simple Namespace

Send feedback on this topic to the SourceAFIS maintainer.
Get/set DPI for processed images.

**Namespace:**  [SourceAFIS.Simple](#)

**Assembly:**  SourceAFIS (in SourceAFIS.dll)
**Syntax**

**Visual Basic (Declaration)**

```vbnet
Public Property Dpi As Integer
    Get
    Set
```

**C#**

```csharp
public int Dpi { get; set; }
```

**Visual C++**

```cpp
public:
    property int Dpi {
        int get ();
        void set (int value);
    }
```

**JavaScript**

```javascript
function get_dpi();
function set_dpi(value);
```

**Field Value**

DPI of images submitted for template extraction. Default is 500.
Remarks

DPI of common optical fingerprint readers is 500. For other types of readers as well as for high-resolution readers, you might need to change this property to reflect capabilities of your reader. This value is used only during template extraction (Extract(Person)). Matching is not affected, because extraction process rescales all templates to 500dpi internally.

Setting DPI causes extractor to adjust its parameters to the DPI. It therefore helps with accuracy. Correct DPI also allows matching of fingerprints coming from different readers. When matching children's fingerprints, it is sometimes useful to fool the extractor with lower DPI setting to deal with the tiny ridges on fingers of children.
See Also

AfisEngine Class
SourceAFIS.Simple Namespace
AfisEngine.Extract(Person)

Send feedback on this topic to the SourceAFIS maintainer.
Get/set minimum number of fingerprints that must match in order for a whole person to match.

**Namespace:**  [SourceAFIS.Simple](#)  
**Assembly:**  SourceAFIS (in SourceAFIS.dll)
Syntax

Visual Basic (Declaration)

Public Property MinMatches As Integer
    Get
    Set

C#

public int MinMatches { get; set; }

Visual C++

public:
    property int MinMatches {
        int get ();
        void set (int value);
    }

JavaScript

function get_minMatches();
function set_minMatches(value);

Field Value

Number of fingerprints that must match during multi-finger matching. Default value is 1 (person matches if any of its fingerprints matches).
Remarks

When there are multiple Fingerprints per Person, SourceAFIS compares every probe Fingerprint to every candidate Fingerprint and takes the best match, the one with highest similarity score. This behavior improves FRR (false reject rate), because low similarity scores caused by damaged fingerprints are ignored. This happens when MinMatches is 1 (default).

When MinMatches is 2 or higher, SourceAFIS compares every probe Fingerprint to every candidate Fingerprint and records score for every comparison. It then sorts collected partial scores in descending order and picks score that is on position specified by MinMatches property, e.g. 2nd score if MinMatches is 2, 3rd score if MinMatches is 3, etc. When combined with Threshold, this property essentially specifies how many partial scores must be above Threshold in order for the whole Person to match. As a special case, when there are too few partial scores (less than value of MinMatches), SourceAFIS picks the lowest score.

MinMatches is useful in some rare cases where there is significant risk that some fingerprint might match randomly with high score due to a broken template or due to some rarely occurring matcher flaw. In these cases, MinMatches might improve FAR. This is discouraged practice though. Application developers seeking ways to improve FAR would do much better to increase Threshold. Threshold can be safely raised to levels where FAR is essentially zero as far as fingerprints are of good quality.
See Also

AfisEngine Class
SourceAFIS.Simple Namespace
AfisEngine:::Verify(Person, Person)
AfisEngine:::Identify(Person, IEnumerable<Of <<(Person)>>))
AfisEngine:::Threshold

Send feedback on this topic to the SourceAFIS maintainer.
Get/set similarity score threshold.

**Namespace:** SourceAFIS.Simple

**Assembly:** SourceAFIS (in SourceAFIS.dll)
**Syntax**

**Visual Basic (Declaration)**

Public Property Threshold As Single
Get
Set

**C#**

public float Threshold { get; set; }

**Visual C++**

public: float Threshold {
    float get ();
    void set (float value);
}

**JavaScript**

function get_threshold();
function set_threshold(value);

**Field Value**

Similarity score threshold for making match/non-match decisions. Default value is rather arbitrarily set to 25.
Remarks

Matching algorithm produces similarity score which is a measure of similarity between two fingerprints. Applications however need clear match/non-match decisions. Threshold is used to turn similarity score into match/non-match decision. Similarity score at or above Threshold is considered match. Lower score is considered non-match. This property is used by `Verify(Person, Person)` and `Identify(Person, IEnumerable<Of <<(Person)>>)` methods to make match decisions.

Appropriate Threshold is application-specific. Application developer must adjust this property to reflect differences in fingerprint readers, population, and application requirements. Start with default threshold. If there are too many false accepts (SourceAFIS reports match for fingerprints from two different people), increase the Threshold. If there are too many false rejects (SourceAFIS reports non-match for two fingerprints of the same person), decrease the Threshold. Every application eventually arrives at some reasonable balance between FAR (false accept ratio) and FRR (false reject ratio).
See Also

AfisEngine Class
SourceAFIS.Simple Namespace
AfisEngine..::.Verify(Person, Person)
AfisEngine..::.Identify(Person, IEnumerable(Of <<(Person)>>))

Send feedback on this topic to the SourceAFIS maintainer.
Finger position on hand.

**Namespace:**  [SourceAFIS.Simple](#)  
**Assembly:**  SourceAFIS (in SourceAFIS.dll)
Syntax

Visual Basic (Declaration)

<SerializableAttribute>_Public Enumeration Finger

C#

[SerializableAttribute]
public enum Finger

Visual C++

[SerializableAttribute]
public enum class Finger

JavaScript

SourceAFIS.Simple.Finger = function();
SourceAFIS.Simple.Finger.createEnum('SourceAFIS.Simple.Finger', false)
<table>
<thead>
<tr>
<th>Member name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Any</td>
<td>Unspecified finger position.</td>
</tr>
<tr>
<td>RightThumb</td>
<td>Thumb finger on the right hand.</td>
</tr>
<tr>
<td>LeftThumb</td>
<td>Thumb finger on the left hand.</td>
</tr>
<tr>
<td>RightIndex</td>
<td>Index finger on the right hand.</td>
</tr>
<tr>
<td>LeftIndex</td>
<td>Index finger on the left hand.</td>
</tr>
<tr>
<td>RightMiddle</td>
<td>Middle finger on the right hand.</td>
</tr>
<tr>
<td>LeftMiddle</td>
<td>Middle finger on the left hand.</td>
</tr>
<tr>
<td>RightRing</td>
<td>Ring finger on the right hand.</td>
</tr>
<tr>
<td>LeftRing</td>
<td>Ring finger on the left hand.</td>
</tr>
<tr>
<td>RightLittle</td>
<td>Little finger on the right hand.</td>
</tr>
<tr>
<td>LeftLittle</td>
<td>Little finger on the left hand.</td>
</tr>
</tbody>
</table>
Remarks

Finger position is used to speed up matching by skipping fingerprint pairs that cannot match due to incompatible position. SourceAFIS will return zero similarity score for incompatible fingerprint pairs.

This feature is optional. It can be disabled by using finger position Any which is default value of Finger for new Fingerprint objects.

A compatible fingerprint pair consists of two fingerprints with the same finger position, e.g. RightThumb matches only other RightThumb. Alternatively, compatible fingerprint pair can be also formed if one of the fingerprints has Any finger position, e.g. Any can be matched against all other finger positions and all other finger positions can be matched against Any. Two fingerprints with Any positions are compatible as well, of course.
See Also

SourceAFIS.Simple Namespace
Fingerprint,

Send feedback on this topic to the SourceAFIS maintainer.
SourceAFIS Fingerprint Recognition System
Fingerprint Class

Collection of fingerprint-related information.

Namespace:  SourceAFIS.Simple
Assembly:  SourceAFIS (in SourceAFIS.dll)
Syntax

Visual Basic (Declaration)

<SerializableAttribute> _
Public Class Fingerprint

C#

[SerializableAttribute]
public class Fingerprint

Visual C++

[SerializableAttribute]
public ref class Fingerprint

JavaScript

SourceAFIS.Simple.Fingerprint = function();

Type.createClass(
    'SourceAFIS.Simple.Fingerprint');
Remarks

This class contains basic information (Image, Template) about the fingerprint that is used by SourceAFIS to perform template extraction and fingerprint matching. If you need to attach application-specific information to Fingerprint object, inherit from this class and add fields as necessary. Fingerprint objects can be grouped in Person objects.

This class is designed to be easy to serialize in order to be stored in binary format (BLOB) in application database, binary or XML files, or sent over network. You can either serialize the whole object or serialize individual properties. You can set some properties to null (Nothing in Visual Basic) to exclude them from serialization.
Inheritance Hierarchy

System...Object
SourceAFIS.Simple...Fingerprint
See Also

Fingerprint Members
SourceAFIS.Simple Namespace
SourceAFIS.Simple..Person

Send feedback on this topic to the SourceAFIS maintainer.
The **Fingerprint** type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fingerprint</td>
<td>Creates empty Fingerprint object.</td>
</tr>
</tbody>
</table>
# Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clone</td>
<td>Create deep copy of the Fingerprint.</td>
</tr>
</tbody>
</table>
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AsBitmap</td>
<td>Fingerprint image as <code>Bitmap</code> object.</td>
</tr>
<tr>
<td>AsBitmapSource</td>
<td>Fingerprint image as <code>BitmapSource</code> object.</td>
</tr>
<tr>
<td>AsImageData</td>
<td>Fingerprint image as raw image in byte array.</td>
</tr>
<tr>
<td>Image</td>
<td>Fingerprint image.</td>
</tr>
<tr>
<td>AsXmlTemplate</td>
<td>Fingerprint template in readable XML format.</td>
</tr>
<tr>
<td>AsIsoTemplate</td>
<td>Fingerprint template in standard ISO format.</td>
</tr>
<tr>
<td>Template</td>
<td>Fingerprint template.</td>
</tr>
<tr>
<td>Finger</td>
<td>Position of the finger on hand.</td>
</tr>
</tbody>
</table>
See Also

Fingerprint Class
SourceAFIS.Simple Namespace

Send feedback on this topic to the SourceAFIS maintainer.
Fingerprint Constructor

Creates empty `Fingerprint` object.

**Namespace:**  `SourceAFIS.Simple`  
**Assembly:**  `SourceAFIS` (in SourceAFIS.dll)
Syntax

Visual Basic (Declaration)

Public Sub New

C#

public Fingerprint()

Visual C++

public:
Fingerprint()

JavaScript

SourceAFIS.Simple.Fingerprint = function();
See Also

Fingerprint Class
SourceAFIS.Simple Namespace

Send feedback on this topic to the SourceAFIS maintainer.
The `Fingerprint` type exposes the following members.
# Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clone</td>
<td>Create deep copy of the Fingerprint.</td>
</tr>
</tbody>
</table>
See Also

Fingerprint Class
SourceAFIS.Simple Namespace

Send feedback on this topic to the SourceAFIS maintainer.
Create deep copy of the Fingerprint.

**Namespace:**  SourceAFIS.Simple

**Assembly:**  SourceAFIS (in SourceAFIS.dll)
Syntax

Visual Basic (Declaration)

Public Function Clone As Fingerprint

C#

public Fingerprint Clone()

Visual C++

public: Fingerprint^ Clone()

JavaScript

function clone();

Return Value

Deep copy of this Fingerprint.
See Also

Fingerprint Class
SourceAFIS.Simple Namespace

Send feedback on this topic to the SourceAFIS maintainer.
The Fingerprint type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AsBitmap</td>
<td>Fingerprint image as <code>Bitmap</code> object.</td>
</tr>
<tr>
<td>AsBitmapSource</td>
<td>Fingerprint image as <code>BitmapSource</code> object.</td>
</tr>
<tr>
<td>AsImageData</td>
<td>Fingerprint image as raw image in byte array.</td>
</tr>
<tr>
<td>Image</td>
<td>Fingerprint image.</td>
</tr>
<tr>
<td>AsXmlTemplate</td>
<td>Fingerprint template in readable XML format.</td>
</tr>
<tr>
<td>AsIsoTemplate</td>
<td>Fingerprint template in standard ISO format.</td>
</tr>
<tr>
<td>Template</td>
<td>Fingerprint template.</td>
</tr>
<tr>
<td>Finger</td>
<td>Position of the finger on hand.</td>
</tr>
</tbody>
</table>
See Also

Fingerprint Class
SourceAFIS.Simple Namespace

Send feedback on this topic to the SourceAFIS maintainer.
Fingerprint image as Bitmap object.

**Namespace:** SourceAFIS.Simple
**Assembly:** SourceAFIS (in SourceAFIS.dll)
Syntax

Visual Basic (Declaration)

Public Property AsBitmap As Bitmap
    Get
    Set

C#

public Bitmap AsBitmap { get; set; }

Visual C++

public: Bitmap^ AsBitmap { 
    Bitmap^ get ();
    void set (Bitmap^ value);
}

JavaScript

function get_asBitmap();
function set_asBitmap(value);

Field Value

Fingerprint image from Image property converted to Bitmap object or nullNothingnullptr a null reference (Nothing in Visual Basic) if Image is nullNothingnullptr a null reference (Nothing in Visual Basic).
Remarks

Use this property in WinForms applications.
See Also

Fingerprint Class
SourceAFIS.Simple Namespace
Fingerprint:::Image
Fingerprint:::AsImageData
Fingerprint:::AsBitmapSource
Fingerprint:::Template
AfisEngine:::Extract(Person)

Send feedback on this topic to the SourceAFIS maintainer.
Fingerprint image as **BitmapSource** object.

**Namespace:**  **SourceAFIS.Simple**  
**Assembly:**  **SourceAFIS (in SourceAFIS.dll)**
Syntax

**Visual Basic (Declaration)**

Public Property AsBitmapSource As BitmapSource
  Get
  Set

**C#**

public BitmapSource AsBitmapSource { get; set; }

**Visual C++**

public: property BitmapSource^ AsBitmapSource {
  BitmapSource^ get ();
  void set (BitmapSource^ value);
}

**JavaScript**

function get_asBitmapSource();
function set_asBitmapSource(value);

**Field Value**

Fingerprint image from Image property converted to BitmapSource object or null (Nothing in Visual Basic) if Image is null (Nothing in Visual Basic).
Remarks

Use this property in WPF applications.
See Also

Fingerprint Class
SourceAFIS.Simple Namespace
Fingerprint::Image
Fingerprint::AsImageData
Fingerprint::AsBitmap
Fingerprint::Template
AfisEngine::Extract(Person)

Send feedback on this topic to the SourceAFIS maintainer.
Fingerprint image as raw image in byte array.

**Namespace:**  SourceAFIS.Simple

**Assembly:**  SourceAFIS (in SourceAFIS.dll)
Syntax

Visual Basic (Declaration)

Public Property AsImageData As Byte()
    Get
    Set

C#

public byte[] AsImageData { get; set; }

Visual C++

public:
property array<unsigned char>^ AsImageData {
    array<unsigned char>^ get();
    void set (array<unsigned char>^ value);
}

JavaScript

function get_asImageData();
function set_asImageData(value);

Field Value

Fingerprint image from Image property converted to raw image (one-dimensional byte array) or nullNothingnullptr null reference (Nothing in Visual Basic) if Image is nullNothingnullptr null reference (Nothing in Visual Basic).
See Also

Fingerprint Class
SourceAFIS.Simple Namespace
Fingerprint..Image
Fingerprint..AsBitmap
Fingerprint..AsBitmapSource
Fingerprint..Template
AfisEngine..Extract(Person)

Send feedback on this topic to the SourceAFIS maintainer.
Fingerprint template in standard ISO format.

**Namespace:** [SourceAFIS.Simple](#)  
**Assembly:** SourceAFIS (in SourceAFIS.dll)
Syntax

Visual Basic (Declaration)

Public Property AsIsoTemplate As Byte()
    Get
    Set

C#

public byte[] AsIsoTemplate { get; set; }

Visual C++

public:
    property array<unsigned char>^ AsIsoTemplate {
        array<unsigned char>^ get ();
        void set (array<unsigned char>^ value);
    }

JavaScript

function get_asIsoTemplate();
function set_asIsoTemplate(value);

Field Value

Remarks

Use this property for two-way exchange of fingerprint templates with other biometric systems. For general use in SourceAFIS, use Template property which contains native template that is fine-tuned for best accuracy and performance in SourceAFIS.

SourceAFIS contains partial implementation of ISO/IEC 19794-2 (2005) standard. Multi-fingerprint ISO templates must be split into individual fingerprints before they are used in SourceAFIS. Value of Finger property is not automatically stored in the ISO template. It must be decoded separately.
See Also

Fingerprint Class
SourceAFIS.Simple Namespace
Fingerprint..Template
Fingerprint..AsXmITemplate
AfisEngine..Extract(Person)

Send feedback on this topic to the SourceAFIS maintainer.
Fingerprint template in readable XML format.

**Namespace:**  [SourceAFIS.Simple](#)

**Assembly:**  SourceAFIS (in SourceAFIS.dll)
Syntax

Visual Basic (Declaration)

Public Property AsXmlTemplate As XElement
    Get
        ...
    Set
        ...

C#

public XElement AsXmlTemplate { get; set; }

Visual C++

public:
    property XElement^ AsXmlTemplate {
        XElement^ get ();
        void set (XElement^ value);
    }

JavaScript

function get_asXmlTemplate();
function set_asXmlTemplate(value);

Field Value

Value of Template converted to SourceAFIS XML template format. This property is nullNothingnullptr null reference (Nothing in Visual Basic) if Template is nullNothingnullptr null reference (Nothing in Visual Basic).
Remarks

Use XML template format where clean data format is more important than compact and fast encoding. XML templates are suitable for XML-based data exchange, encoding of multiple fingerprints along with accompanying data into single XML file, and for debugging and logging purposes.

Value of Finger property is not automatically stored in the XML template. It must be decoded separately.
See Also

Fingerprint Class
SourceAFIS.Simple Namespace
Fingerprint::Template
Fingerprint::AsIsoTemplate
AfisEngine::Extract(Person)

Send feedback on this topic to the SourceAFIS maintainer.
Position of the finger on hand.

**Namespace:** [SourceAFIS.Simple](#)  
**Assembly:** SourceAFIS (in SourceAFIS.dll)
Syntax

Visual Basic (Declaration)

Public Property Finger As Finger
    Get
    Set

C#

public Finger Finger { get; set; }

Visual C++

public:
property Finger Finger {
    Finger get ();
    void set (Finger value);
}

JavaScript

function get_finger();
function set_finger(value);

Field Value

Finger (thumb to little) and hand (right or left) that was used to create this fingerprint. Default value Any means unspecified finger position.
Remarks

Finger position is used to speed up matching by skipping fingerprint pairs with incompatible finger positions. Check Finger enumeration for information on how to control this process. Default value Any disables this behavior.
See Also

Fingerprint Class
SourceAFIS.Simple Namespace
SourceAFIS.Simple..::..Finger

Send feedback on this topic to the SourceAFIS maintainer.
Fingerprint image.

**Namespace:**  [SourceAFIS.Simple](SourceAFIS.Simple)

**Assembly:**  SourceAFIS (in SourceAFIS.dll)
Syntax

Visual Basic (Declaration)

Public Property Image As Byte(),
        Get
        Set

C#

class	public	byte[,] Image { get; set; }

Visual C++

public:
        property array<unsigned char,2>^ Image {
            array<unsigned char,2>^ get ();
            void set (array<unsigned char,2>^ value);
        }

JavaScript

function get_image();
function set_image(value);

Field Value

Raw fingerprint image that was used to extract the Template or other image attached later after extraction. This property is nullNothingnullptr or null reference (Nothing in Visual Basic) by default.
Remarks

This is the fingerprint image. This property must be set before call to `Extract(Person)` in order to generate valid `Template`. Once the `Template` is generated, Image property has only informational meaning and it can be set to `null/Nothing/nullptr` null reference (Nothing in Visual Basic) to save space. It is however recommended to keep the original image just in case it is needed to regenerate the `Template` in future.

The format of this image is a simple raw 2D array of bytes. Every byte represents shade of gray from black (0) to white (255). When indexing the 2D array, Y axis goes first, X axis goes second, e.g. Image[y, x]. To convert to/from `Bitmap` object, use `AsBitmap` property. To convert to/from `BitmapSource` object, use `AsBitmapSource` property.

Accessors of this property do not clone the image. To avoid unwanted sharing of the byte array, call `Clone()` on the Image.
See Also

Fingerprint Class
SourceAFIS.Simple Namespace
Fingerprint:::Template
Fingerprint:::AsBitmap
Fingerprint:::AsBitmapSource
Fingerprint:::AsImageData
AfisEngine:::Extract(Person)

Send feedback on this topic to the SourceAFIS maintainer.
Fingerprint template.

**Namespace:** [SourceAFIS.Simple](#)

**Assembly:** SourceAFIS (in SourceAFIS.dll)
Syntax

Visual Basic (Declaration)

Public Property Template As Byte()
   Get
   Set

C#

public byte[] Template { get; set; }

Visual C++

public:
property array<unsigned char>^ Template {
   array<unsigned char>^ get ()
   void set (array<unsigned char>^ value);
}

JavaScript

function get_template();
function set_template(value);

Field Value

Fingerprint template generated by Extract(Person) or other template assigned for example after deserialization. This property is nullNothingnullptr a null reference (Nothing in Visual Basic) by default.
Remarks

Fingerprint template is an abstract model of the fingerprint that is serialized in a very compact binary format (up to a few KB). Templates are better than fingerprint images, because they require less space and they are easier to match than images. To generate Template, pass `Fingerprint` object with valid `Image` to `Extract(Person)`. Template is required by `Verify(Person, Person)` and `Identify(Person, IEnumerable<Of <='(Person)>>)`.

Format of the template may change in later versions of SourceAFIS. Applications are recommended to keep the original `Image` in order to be able to regenerate the Template.
See Also

Fingerprint Class
SourceAFIS.Simple Namespace
Fingerprint::{:Image
AfisEngine::{:Extract(Person)
Fingerprint::{:AsIsoTemplate
Fingerprint::{:AsXmlTemplate

Send feedback on this topic to the SourceAFIS maintainer.
Collection of *Fingerprint* belonging to one person.

**Namespace:**  [SourceAFIS.Simple](#)  
**Assembly:**  SourceAFIS (in SourceAFIS.dll)
Syntax

Visual Basic (Declaration)

```vbnet
<SerializableAttribute> _
Public Class Person
```

C#

```csharp
[SerializableAttribute]
public class Person
```

Visual C++

```cpp
[SerializableAttribute]
pUBLIC ref class Person
```

JavaScript

```javascript
SourceAFIS.Simple.Person = function();
Type.createClass(
    'SourceAFIS.Simple.Person');
```
Remarks

This class is primarily a way to group multiple Fingerprints belonging to one person. This is very convenient feature when there are multiple fingerprints per person, because it is possible to match two Persons directly instead of iterating over their Fingerprint.

Id property is provided as a simple means to bind Person objects to application-specific information. If you need more flexibility, inherit from Person class and add application-specific fields as necessary.

This class is designed to be easy to serialize in order to be stored in binary format (BLOB) in application database, binary or XML files, or sent over network. You can either serialize the whole Person or serialize individual Fingerprint.
Inheritance Hierarchy

System..::..Object
SourceAFIS.Simple..::..Person
See Also

Person Members
SourceAFIS.Simple Namespace
SourceAFIS.Simple.Fingerprint

Send feedback on this topic to the SourceAFIS maintainer.
The **Person** type exposes the following members.
## Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Person</td>
<td>Overloaded.</td>
</tr>
</tbody>
</table>
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clone</td>
<td>Create deep copy of the Person.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Id</td>
<td>Application-assigned ID for the Person.</td>
</tr>
<tr>
<td>Fingerprint</td>
<td>List of Fingerprint belonging to the Person.</td>
</tr>
</tbody>
</table>
See Also

Person Class
SourceAFIS.Simple Namespace

Send feedback on this topic to the SourceAFIS maintainer.
Visual Basic  C#
Visual C++  JavaScript
Include Protected Members
Include Inherited Members
SourceAFIS Fingerprint Recognition System
Person Constructor

Person Class  See Also  Send Feedback
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Person()()()</td>
<td>Creates an empty <code>Person</code> object.</td>
</tr>
<tr>
<td>Person(array&lt;Fingerprint&gt;[],[],[])</td>
<td>Creates new <code>Person</code> object and initializes it with a list of Fingerprint objects.</td>
</tr>
</tbody>
</table>
See Also

Person Class
Person Members
SourceAFIS.Simple Namespace

Send feedback on this topic to the SourceAFIS maintainer.
Person Constructor

Creates an empty Person object.

Namespace: SourceAFIS.Simple
Assembly: SourceAFIS (in SourceAFIS.dll)
Syntax

Visual Basic (Declaration)

Public Sub New

C#

public Person()

Visual C++

public:
Person()

JavaScript

SourceAFIS.Simple.Person = function();
See Also

Person Class
Person Overload
SourceAFIS.Simple Namespace

Send feedback on this topic to the SourceAFIS maintainer.
Person Constructor (array<Fingerprint>[][])

Creates new Person object and initializes it with a list of Fingerprints.

Namespace: SourceAFIS.Simple
Assembly: SourceAFIS (in SourceAFIS.dll)
## Syntax

### Visual Basic (Declaration)

```vbnet
Public Sub New ( _
    ParamArray fingerprints As Fingerprint() _
)
```

### C#

```csharp
public Person(
    params Fingerprint[] fingerprints
)
```

### Visual C++

```cpp
public: 
Person(
    ... array<Fingerprint>^ fingerprints
)
```

### JavaScript

```javascript
SourceAFIS.Simple.Person = function(... fingerprints);
```

## Parameters

- **fingerprints**
  - Type: array<SourceAFIS.Simple...Fingerprint>[][]
  - `Fingerprint` objects to add to the new `Person`.
See Also

Person Class
Person Overload
SourceAFIS.Simple Namespace

Send feedback on this topic to the SourceAFIS maintainer.
The **Person** type exposes the following members.
# Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clone</td>
<td>Create deep copy of the Person.</td>
</tr>
</tbody>
</table>
See Also

Person Class
SourceAFIS.Simple Namespace
Send feedback on this topic to the SourceAFIS maintainer.
Create deep copy of the `Person`.

**Namespace:**  [SourceAFIS.Simple](#)

**Assembly:**  SourceAFIS (in SourceAFIS.dll)
Syntax

Visual Basic (Declaration)

Public Function Clone As Person

C#

public Person Clone()

Visual C++

public:
Person^ Clone()

JavaScript

function clone();

ReturnValue

Deep copy of the Person.
Remarks

This method also clones all Fingerprint objects contained in this Person.
See Also

Person Class
SourceAFIS.Simple Namespace

Send feedback on this topic to the SourceAFIS maintainer.
The `Person` type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Id</td>
<td>Application-assigned ID for the Person.</td>
</tr>
<tr>
<td>Fingerprint</td>
<td>List of Fingerprint belonging to the Person.</td>
</tr>
</tbody>
</table>
See Also

Person Class
SourceAFIS.Simple Namespace

Send feedback on this topic to the SourceAFIS maintainer.
SourceAFIS Fingerprint Recognition System

Person..::.Fingerprints Property

Namespace:  SourceAFIS.Simple
Assembly:  SourceAFIS (in SourceAFIS.dll)
Syntax

Visual Basic (Declaration)

Public Property Fingerprints As List(Of Fingerprint)
    Get
    Set

C#

public List<Fingerprint> Fingerprints { get; set; }

Visual C++

public:
    property List<Fingerprint>^ Fingerprints {
        List<Fingerprint>^ get ();
        void set (List<Fingerprint>^ value);
    }

JavaScript

function get_fingerprints();
function set_fingerprints(value);
Remarks

This collection is initially empty. Add Fingerprint objects here. You can also assign the whole collection.
See Also

Person Class
SourceAFIS.Simple Namespace
SourceAFIS.Simple::<::Fingerprint

Send feedback on this topic to the SourceAFIS maintainer.
SourceAFIS Fingerprint Recognition System

**Person.Id Property**

Application-assigned ID for the [Person](#).

**Namespace:** [SourceAFIS.Simple](#)

**Assembly:** SourceAFIS (in SourceAFIS.dll)
Syntax

Visual Basic (Declaration)

Public Property Id As Integer
    Get
    Set

C#

public int Id { get; set; }

Visual C++

public:
    property int Id {
        int get ();
        void set (int value);
    }

JavaScript

function get_id();
function set_id(value);
Remarks

SourceAFIS doesn't use this property. It is provided for applications as an easy means to link Person objects back to application-specific data. Applications can store any integer ID in this field, for example database table key or an array index.

Applications that need to attach more detailed information to the person should inherit from Person class and add fields as necessary.
See Also

Person Class
SourceAFIS.Simple Namespace

Send feedback on this topic to the SourceAFIS maintainer.