Light weight Framework for RFID Applications

This white paper elucidates the development of a Framework for RFID applications, which acts as an interface between RFID Controls like tags/readers/controls obtained from different RFID vendors and the RFID based applications like ERP. The Framework seamlessly integrates with RFID control and with legacy or modern databases. The customers can switch from one RFID vendor to other without modifying the source code of the existing application.
Introduction

Radio Frequency Identification (RFID) is a sight less identification system. It is the combination of Radio Frequency and the Microchip technologies. RFID transmits the identity of an object as a unique serial number. This identity is stored in the tag chips and can be retrieved by the Readers. The components of RFID are

- Tag
- Reader

Tag

Tag, also called transponder is a small device which contains a microchip. The chip is used to store the data. The tag can be programmed with specific items of information, such as an ID or serial number or a user data. The tag identifies itself by transmitting signals to the reader. There are two types of Tags namely,

- Active Tag
- Passive TaG

Active Tag

- Transmits signals from the micro chip circuit through the power obtained from an internal battery
- High Signal range
- Used on large assets, such as cargo containers, rail cars, and large reusable containers
- Costlier and Larger in Size

Passive Tag

- Contains no power source
- Obtains power from the reader
- Low Signal range
- Cheaper and Smaller than active tags

Reader

A reader consists of one or more antennas that emit and receive radio waves. The micro chip circuit present in the tags is powered by this signal. When the tags enters into the radio wave field, it transmits its unique information (unique serial number or user data) to the reader by modulating the signal. The reader converts the signal obtained from the tag as digital information and passes to the applications.

Code DOM Emit

Code Document Object Model (Code DOM) is a helps in creating, compiling and executing a programming structure dynamically at runtime. This programming structure involves Namespace, Type – Class, Interface, Fields, Properties, Parameters and even methods. It represents source code trees in memory. Each source code tree is called a CompileUnit. CompileUnits are linked together to form a tree structure in memory representing the structure of the source code. The ICodeGenerator Interface is what receives the Compile Units and outputs source code in the designated language. Source code may be generated for any language.
that supports the Code DOM specification. Code DOM allows dynamically compiling and creating assemblies also.

The main purpose of Code DOM is to minimize repetitive coding tasks, and to minimize the number of human-generated source code lines being created. The Code DOM provides a language independent object model for representing the structure of source code in memory.

**XML**

Extensible Markup Language (XML) is a meta-language for describing markup languages. Markup language is a mechanism to identify structures in a document. XML uses Schema to define the structure of an XML document and to describe the data present in it.

XML Schema of RFID Framework

```xml
<!—XML Schema Definition—>
<xsd:schema xmlns:xsd="......">
  targetNamespace="RFID_Framework"<!-- Namespace -->
  <xs:element name = RFID_Connection><!-- Class Name -->
    <xs:complexType>
      <xs:sequence>
        <xs:element name = "RFID_Port_Number" type="xs:string"/><!-- Properties of Class -->
        <xs:element name = "RFID_Port_Invoke" type="xs:string"/><!-- Properties of Class -->
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xsd:schema>
```

**RFID Control**

RFID Control is the one which is supplied by the RFID Component vendor. The Control can be an ActiveX Control (OCX) or the .NET assembly (dll). The RFID Control supplied by the different vendors will have certain common characteristics like:

**Reader Properties**

- Reader ID
- Tag ID
- Gain Mode
- Site Code
- Tag Time out value
- RSSI Value
- Port to Communicate with the RFID Control

**Reader Events**

- Event to trigger when a new tag arrives
- Event to handle continuous signals from tags
- Event to trigger when a Tag is Time out.
**A Comparison with RFID and Barcode**

<table>
<thead>
<tr>
<th>Feature</th>
<th>RFID</th>
<th>Barcode</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ability to read more than one item at a time</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Read the item while moving</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Line of Site Read not Required</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Built in Security</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Environment Resistance</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Number or Reads</td>
<td>More</td>
<td>Less</td>
</tr>
</tbody>
</table>

**Existing System**

The RFID based applications and the RFID components are tightly coupled. If the Customer wants to switch from one RFID vendor to another, immense effort is needed for the developer to change the source code of the existing application and to recompile it. This consumes more time and cost.

**RFID Framework**

The RFID Framework generates an assembly which consist common methods and events for different RFID Controls.

**RFID FRAMEWORK**

- The RFID Framework is the middleware between the RFID Hardware\Control and RFID based applications.
- The framework will read the metadata of the assembly or the dll supplied by the vendor, identifies the elements and generates a XML Schema dynamically.
- Using the XML schema, the framework will produce the common methods, events and properties which can be used in any RFID based applications.
- With this ability, the Framework can be used with any kind of application which implements the RFID Technology irrespective of the RFID controls used.
5. For an example, a vendor X provides the control and uses the method name NewTagFound to capture the new tag and vendor Y uses the method name TagIdentified for the same scenario then the framework will identify these methods and produces a constant output method name called OnNewTag. The application can use this method to handle the new tag received irrespective of the RFID Control provided by different vendors.

The Events exposed by the Framework are:
- OnNewTag — When a new tag arrives
- OnTagTransmission — When a tag emitting continuous signal
- OnTagTimeout — When a tag is time out
- OnNewReader — When a new reader is found in the network

The Properties exposed by the framework are:
- SiteCode — Filtering the Tag in the Reader
- ReaderID — Unique ID for the Reader
- ReaderName — Name of the Reader

Developers can intended these functions without concerning about the properties and events name exposed by the RFID Control.

**How Framework works**
- The RFID Control (OCX/DLL) from the RFID vendor is taken as input.
- Framework loads the control into memory, and fetch the elements like Classes, Events, Methods and Properties available in the RFID Control.
Generates a XML Schema, based on the elements obtained using WriteXMLSchema method of Dataset.

Code DOM feature in .NET is used to convert the XML Schema into a Class file which contains the elements.

During the Code DOM conversion, the elements present in the schema will be converted to the common methods and events.

The Class File is then converted to .NET Assembly using the Create Compiler in Code DOM.

The generated Assembly is used in the many several RFID Applications to provide the functionality.

Creation of Class file from XML Schema using Code DOM

<table>
<thead>
<tr>
<th>Layer 1</th>
<th>Device Management</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reader-A</td>
<td>Tag</td>
</tr>
<tr>
<td>Reader-B</td>
<td>Tag</td>
</tr>
<tr>
<td>Reader-C</td>
<td>Tag</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Layer 2</th>
<th>XML Schema Generator</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generates XML Schema using Dataset</td>
<td></td>
</tr>
<tr>
<td>Fetch Elements, Methods and Properties present in the OCX Control</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Layer 3</th>
<th>Class File Generator</th>
</tr>
</thead>
<tbody>
<tr>
<td>Creates Class Files using Code DOM</td>
<td></td>
</tr>
<tr>
<td>Fetch Elements in the XML Schema</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Layer 4</th>
<th>DLL Generator</th>
</tr>
</thead>
<tbody>
<tr>
<td>Complies the Class file and generates the Framework DLL</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Layer 5</th>
<th>Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERP</td>
<td>CRM</td>
</tr>
</tbody>
</table>

Architecture of RFID Framework

- The XML Schema generated using XML Generator is taken as the input.
- The declaration type of the Class and Structure is identified using the CodeTypeDeclaration class.
- Using the CodeTypeMember class the base class and type of the methods, properties and constructors are identified.
- The Methods, Properties and Events are created in the class file using the CodeMemberMethod, CodeMemberProperty and CodeDelegateCreateExpression respectively.

Creation of Generic Framework Assembly

- The Class File generated by the Class File Generator is taken as the input.
- The Code DOM uses the interface namely ICodeCompiler for invoking the Compiler.
- The CSharpCodeProvider provides access to instances of the C# Code generator and code compiler.
The CreateComplier creates an instance of the C# code compiler which is used to compile the source code of the class file and creates the assembly as the output.

- This assembly can be used in the RFID Applications without having the knowledge about the SDK provided by the vendors.
- Set the reference to the assembly provided by the framework in the RFID based applications; compose code to handle the events and properties provided by the framework.

Light weight Framework for RFID Applications
RFID Based Applications

The applications which use the RFID technology can use this framework to feed the necessary inputs like events, properties and methods which are required for processing.

Some of the applications which can be created using this framework are
- Warehouse Management System (WMS)
- Enterprise Resource Planning (ERP)
- Supply Chain Management (SCM)

Conclusion

The RFID replaces the traditional identification methods like barcode. The RFID framework provides the freedom for developers/customers to switch different RFID vendors at any point of time during the development with minimal effort.

References


About Calsoft Labs

Calsoft Labs provides specialized concept to market Product Engineering services to product and technology companies in select market segments. Our target markets include Automotive, Consumer Electronics, Media, Networking, Storage and Independent Software Vendors (ISVs). Calsoft Labs delivers unmatched business value to its customers through a combination of process excellence, reusable frameworks and technology innovation.

Calsoft Labs is a wholly owned subsidiary of ALTEN. Set up in 1988, ALTEN is a European leader in Engineering and Technology Consulting (ETC) with 15,000 employees in over 14 countries worldwide.