# Whats new in caCORE SDK 4.0

caCORE SDK 4.0 is a major release with many new features available. Some of the newly developed features are to strengthen the infrastructure and others are to support new requirements. The purpose of this section is to highlight the major functionality and performance enhancements and improvements introduced in the caCORE SDK 4.0 release.

# Code Generation

The architecture and the core of the code generation module of the caCORE SDK 4.0 has been completely rewritten. Entire code generation framework now runs from a single configuration file based on the Spring Framework as opposed to individual configuration files used by the previous releases. Some of the visible improvements in the code generation module are highlighted below.

* **Support for Enterprise Architect and ArgoUML**

Previous releases of the caCORE SDK used to support only Enterprise Architect as a tool for UML modeling. With SDK 4.0, users can choose between ArgoUML and Enterprise Architect. The added support for ArgoUML provides users an open source alternative to commercial software like Enterprise Architect.

* **Performance Improvement in Code Generation**

The caCORE SDK 4.0 has significantly improved the performance of the code generation module. Average users should notice the system generation process to be completed in approximately 15% of time of what it used to take with the previous releases of SDK.

* **Support for Validators**

The caCORE SDK code generator now has support for validators. Validators serves the purpose of validating the object model and object relational mapping information before the code generator starts. These validators provide descriptive messages to users which allow users to quickly identify the root cause of the code generation failure.

* **Reduced and Improved Generated Artifacts**

The artifacts generated by caCORE SDK are completely redesigned to suit the needs of the newly redesigned runtime system. Artifacts generated by previous release of SDK were not reusable outside of SDK due to certain dependency on SDK, however artifacts generated by SDK 4.0 can be reused anywhere. For e.g. Java beans generated by SDK had getter methods to connect to the server and they were not simple POJOs which in SDK 4.0 are simple POJO beans. The table given below provides list of all the artifacts that has changed.

|  |  |  |
| --- | --- | --- |
| Artifact | Before SDK 4.0 | SDK 4.0 |
| POJO beans for domain objects (\*.java) | C:\Documents and Settings\patelsat\Local Settings\Temporary Internet Files\Content.IE5\GDUZK5EJ\MCj04325300000[1].png | C:\Documents and Settings\patelsat\Local Settings\Temporary Internet Files\Content.IE5\GDUZK5EJ\MCj04325300000[1].png |
| SDK specific Java beans for domain objects (\*.java) | C:\Documents and Settings\patelsat\Local Settings\Temporary Internet Files\Content.IE5\GDUZK5EJ\MCj04325300000[1].png |  |
| “Impl” classes for Java beans (\*.Impl.\*.java) | C:\Documents and Settings\patelsat\Local Settings\Temporary Internet Files\Content.IE5\GDUZK5EJ\MCj04325300000[1].png |  |
| Web Service beans (\*.ws.\*.java) | C:\Documents and Settings\patelsat\Local Settings\Temporary Internet Files\Content.IE5\GDUZK5EJ\MCj04325300000[1].png |  |
| “Impl” classes for web service beans (\*.ws.impl.\*.java) | C:\Documents and Settings\patelsat\Local Settings\Temporary Internet Files\Content.IE5\GDUZK5EJ\MCj04325300000[1].png |  |
| JUnit test cases for domain objects | C:\Documents and Settings\patelsat\Local Settings\Temporary Internet Files\Content.IE5\GDUZK5EJ\MCj04325300000[1].png |  |
| Hibernate O/R mapping files for domain objects (\*.hbm.xml) | C:\Documents and Settings\patelsat\Local Settings\Temporary Internet Files\Content.IE5\GDUZK5EJ\MCj04325300000[1].png | C:\Documents and Settings\patelsat\Local Settings\Temporary Internet Files\Content.IE5\GDUZK5EJ\MCj04325300000[1].png |
| Hibernate O/R mapping files for “Impl” classes (\*Impl.hbm.xml) | C:\Documents and Settings\patelsat\Local Settings\Temporary Internet Files\Content.IE5\GDUZK5EJ\MCj04325300000[1].png |  |
| Hibernate configuration file (\*.cfg.xml) | C:\Documents and Settings\patelsat\Local Settings\Temporary Internet Files\Content.IE5\GDUZK5EJ\MCj04325300000[1].png | C:\Documents and Settings\patelsat\Local Settings\Temporary Internet Files\Content.IE5\GDUZK5EJ\MCj04325300000[1].png |
| Hibernate cache configuration file (ehcache.xml) | C:\Documents and Settings\patelsat\Local Settings\Temporary Internet Files\Content.IE5\GDUZK5EJ\MCj04325300000[1].png | C:\Documents and Settings\patelsat\Local Settings\Temporary Internet Files\Content.IE5\GDUZK5EJ\MCj04325300000[1].png |
| SDK DAO configuration file (DAOConfig.xml) | C:\Documents and Settings\patelsat\Local Settings\Temporary Internet Files\Content.IE5\GDUZK5EJ\MCj04325300000[1].png |  |
| Domain object list (coreBeans.properties) | C:\Documents and Settings\patelsat\Local Settings\Temporary Internet Files\Content.IE5\GDUZK5EJ\MCj04325300000[1].png |  |
| Association mapping file (roleLookup.properties) | C:\Documents and Settings\patelsat\Local Settings\Temporary Internet Files\Content.IE5\GDUZK5EJ\MCj04325300000[1].png |  |
| XML Schema for domain model (\*.XSD) | C:\Documents and Settings\patelsat\Local Settings\Temporary Internet Files\Content.IE5\GDUZK5EJ\MCj04325300000[1].png | C:\Documents and Settings\patelsat\Local Settings\Temporary Internet Files\Content.IE5\GDUZK5EJ\MCj04325300000[1].png |
| Castor mapping files (xml-mapping.xml, xml-unmapping.xml) | C:\Documents and Settings\patelsat\Local Settings\Temporary Internet Files\Content.IE5\GDUZK5EJ\MCj04325300000[1].png | C:\Documents and Settings\patelsat\Local Settings\Temporary Internet Files\Content.IE5\GDUZK5EJ\MCj04325300000[1].png |
| Web service deployment descriptor (server-config.wsdd) | C:\Documents and Settings\patelsat\Local Settings\Temporary Internet Files\Content.IE5\GDUZK5EJ\MCj04325300000[1].png | C:\Documents and Settings\patelsat\Local Settings\Temporary Internet Files\Content.IE5\GDUZK5EJ\MCj04325300000[1].png |

* **Additional UML Features Supported**

caCORE SDK 4.0 now supports many new UML features in the object model and in the object relational mapping aspect.

**Object Model**

* + ID attribute – Users of the caCORE SDK do not have to name the attribute that maps to the primary key column of the corresponding table for the class as “ID”. Users can now specify the attribute mapping to primary key column using a tag value on the class in the domain model.
  + Primitives support – SDK 4.0 allows users to specify Java’s primitive type for any attribute’s data type. SDK 4.0 interprets these primitives in the wrapper data type during code generation.
  + Collection of primitives –Users of the SDK can now use collection of primitives or wrapper data types as the type of the attribute.

**Object Relational Mapping**

* + Inheritance – caCORE SDK 4.0 now supports an alternate way of mapping inheritance hierarchy in the database. SDK users can choose between existing *Table per class* mechanism to map inheritance in the database or they can choose T*able per inheritance* hierarchy for the mapping.
  + Join tables – Previous releases of the SDK used to support join tables only for the many to many type of associations. With SDK 4.0 users can choose to use join tables for any type of associations.

# Generated System

In addition to the new code generator module, caCORE SDK 4.0 has introduced significant changes in the runtime system. Since many of the changes are in the infrastructure mostly users utilizing advance options will notice or be affected by the restructuring of the SDK’s runtime system

* **Client Server Infrastructure**
* The client-server infrastructure of SDK used to rely on the Java beans developed specifically for SDK. These specialized java beans had the capability to connect to the server when required to fetch the associated objects. With SDK 4.0, regular POJOs are used in conjunction with concepts from Aspect Oriented Programming (AOP) to facilitate similar mechanism. With this design approach, domain object beans generated by SDK are true POJOs and can be used outside of SDK easily.
* In addition to the restructuring of the Java beans with AOP, SDK 4.0 now also can connect to various SDK generated system from within the same client JVM. In previous versions, users of the SDK could connect to only one remote service at a time; with this feature, developers will now be able to retrieve data from multiple data services.
* **Simplified Application Service**

Many of the existing methods of the ApplicationService interface have been deprecated. Newly added methods have syntax similar to the existing methods but they now require less information. The simplified Application Service will be easier to work with

* **Web Services**
  + SDK 4.0 generated web services work on the simple POJO beans. The web service from previous version of SDK required specialized POJO beans in the .ws package whereas SDK 4.0 generated web services utilizes the same Java beans that are used by the other tiers of the application.
  + SDK 4.0 web services also have additional methods to allow users to fetch the associations of the domain object. Users can now specify which specific association they would like to fetch from the server.
  + Starting with version 4.0, users of the SDK will not have to deploy the web service independently. The SDK 4.0 generated web services are embedded in the .war file and will be deployed automatically when the application server starts
* **Graphical User Interface**
  + The caCORE SDK 4.0 generated system has a newly developed graphical user interface. This new interface allows users a richer experience.
  + Security of the new user interface has been enhanced. Users now have access to built in security capabilities such that when the security is enabled in the system, users will get experience of completely secured system and not just one of the secured interface.
  + The caCORE SDK 4.0 generated GUI now has embedded Javadocs for the domain objects for which the system was originally generated. Users of the web interface can browse the Javadocs by visiting a link on the generated system’s home page
  + Previous release of the SDK did not allow fetching of an associated object that had more than one association with another object. The newly generated web interface allows user to retrieve associations regardless of the number of associations between two objects.
* **Security**
  + The caCORE SDK 4.0 has a completely new security implementation that is based on ACEGI security framework. The previous implementation of security in the caCORE SDK was weaved into the application logic. For caCORE SDK 4.0, security implementation is kept outside of the application and is managed through Aspect Oriented Programming principles. SDK users can now easily change the implementation of security without going into the details of SDK’s code base.
  + Instance level security – The caCORE SDK 4.0 supports instance level security utilizing CSM, which provides flexibility to provide more granular access to the data. For e.g. Users can be given access to only a subset of records from a particular table vs all the records of a particular table.
  + Attribute level security – In addition to the instance level security, the caCORE SDK 4.0 also provides very granular attribute level security to the users. For e.g. Only certain users are allowed to see Social Security Numbers of Person object.
  + Concurrent user access in secured API – Users of the SDK generated java client in the previous releases were constrained to use the same user account throughout the lifecycle of the ApplicationService. In SDK 4.0, users can create many different instances of the ApplicationService and login with different user accounts at the same time from different threads of the client application.