

X9SDK: Your x9.37 development tool

SDK Primary Classes

- X9SdkBase
- X9SdkBase
- X9SdkIo
- X9SdkObject
- X9SdkImage
- X9Object
- X9ObjectManager

Major Support Classes

- X9Binder
- X9Config
- X9Error
- X9ErrorManager
- X9Field
- X9FieldManager
- X9FieldWalker
- X9Mapping
- X9Message
- X9MicrFont
- X9Reader
- X9Repair
- X9Rules
- X9TiffRules
- X9Validator
- X9ValidateHeaders

Major Image Classes

- X9CheckFormat
- X9DrawCheck
- X9ImageCachce
- X9ImageReader
- X9ImageUtilities
- X9Tag
- X9TagManager
- X9Tiff
- X9TiffManager
- X9TiffInspector
- X9TiffReader
- X9TiffRepair

Licensing

- One time charge
- Annual support
- Concurrent licensing of X9Assist is required

X9SDK is our SDK product (software development kit) which provides read/write access and a large number of other functions to x9 files at both the record and field level. SDK classes provide numerous extended functions which are associated with common x9 file processing tasks.

X9SDK is targeted for those organizations that must develop their own applications around the x9 file standards. X9SDK is a proven toolkit that will allow you to quickly develop and test your specific application requirements.

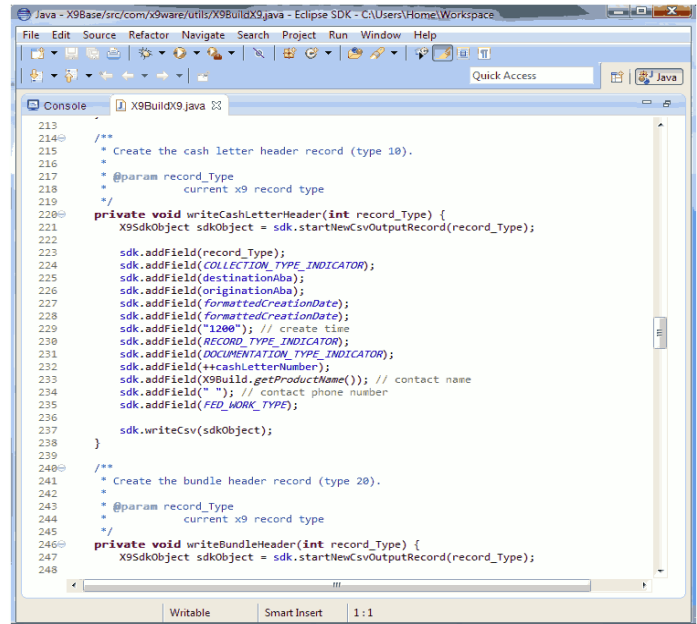
X9SDK is used internally by all of our X9Ware LLC products including X9Assist and X9Utilities. It is a mature product that will continue to be enhanced as our products evolve and as x9 specifications and standard are changed.

X9SDK has all of the x9 development capabilities you are looking for. It supports all of the various x9 standards per our internal XML configuration files. The SDK is 100% Java and hence will run on a variety of software platforms. Our SDK is running on Windows, Linux, and AIX in a variety of environments.

The X9Ware SDK is not an afterthought. It is used internally by X9Assist to perform virtually all x9 related I/O functions. Most of the X9Assist tools are built into the SDK itself, making them available to our SDK users. For example, x9 file validation and tiff image repair are complex functions that are available from the SDK. Consistent use of the SDK within X9Assist simplified the development of our product and we know that it can do the same for you.

X9SDK is the primary Java class that will be created by user programs. An application process should be designed to perform all of its processing using this single SDK instance. Once the SDK instance is created, the desired x9 configuration can be optionally set using Bind Configuration. This assigned configuration indicates which set of x9 rules will be applied to the x9 file, which controls x9 file validation, available record types, field construction, etc. A configuration must be set when x9 files are being created. When reading input x9 files, an alternative is to use X9Binder which will automatically determine the configuration to be used based on the content of the file header record and the user mapping that has been defined. Methods are then provided to take high level actions within the SDK:

- Open and close individual files
- Copy an x9 file to CSV
- Copy a CSV file to x9
- Read and write x9 files record by record
- read and write CSV files record by record
- Write x9 and CSV files field by field
- Get and set images
- Validate x9 data and associated images
- Dynamically draw images from MICR
- Automatically create bundles
- Automatically create trailer records
- Increase performance based on use of concurrent thread processing



```
213
214  /**
215   * Create the cash letter header record (type 10).
216   *
217   * @param record_Type
218   *       current x9 record type
219   */
220  private void writeCashLetterHeader(int record_Type) {
221      X9SdkObject sdkObject = sdk.startNewCsvOutputRecord(record_Type);
222
223      sdk.addField(record_Type);
224      sdk.addField(COLLECTION_TYPE_INDICATOR);
225      sdk.addField(destinationAba);
226      sdk.addField(originationAba);
227      sdk.addField(formattedCreationDate);
228      sdk.addField(formattedCreationDate);
229      sdk.addField("1200"); // create time
230      sdk.addField(RECORD_TYPE_INDICATOR);
231      sdk.addField(DOCUMENTATION_TYPE_INDICATOR);
232      sdk.addField(++cashLetterNumber);
233      sdk.addField(X9Build.getProductNumber());
234      sdk.addField(" "); // contact phone number
235      sdk.addField(FED_MORX_TYPE);
236
237      sdk.writeCsv(sdkObject);
238  }
239
240  /**
241   * Create the bundle header record (type 20).
242   *
243   * @param record_Type
244   *       current x9 record type
245   */
246  private void writeBundleHeader(int record_Type) {
247      X9SdkObject sdkObject = sdk.startNewCsvOutputRecord(record_Type);
248  }
```

X9SdkObject represents a single x9 record within the SDK interface. In addition to x9 record and field data, there are also image based attributes such as the image itself (in one of several forms) and an indicator as to whether the image represents the front or back of the check. When a new x9object is created, the x9 data can be presented in one of several forms. The first option is to read the x9 record from an external source such as x9 or csv. The second option is to present the x9 data within a csv array, which represents the individual fields that are to be ultimately used to create an x9 record. The final option is to dynamically build the x9 data on a record by record and field by field basis. If present, the csv array consists of each field stored in string format, which can consist of either quoted or unquoted strings. This data would typically always be provided in a trimmed format (no leading or trailing spaces), since field justification and padding is automatic based on current x9 rules. Numeric fields can be presented as a simple string without leading zeroes, with required padding automatically inserted.

Methods are then provided to manipulate this data in numerous ways. For example, we can convert the CSV Array to an x9 record, or we can similarly convert the x9 record to a CSV Array. Translations can be done between character sets (for example, EBCDIC to ASCII).

Please visit our website at x9ware.com, or send inquiries to x9assist@x9ware.com. We have the x9 experience to meet your specific needs.