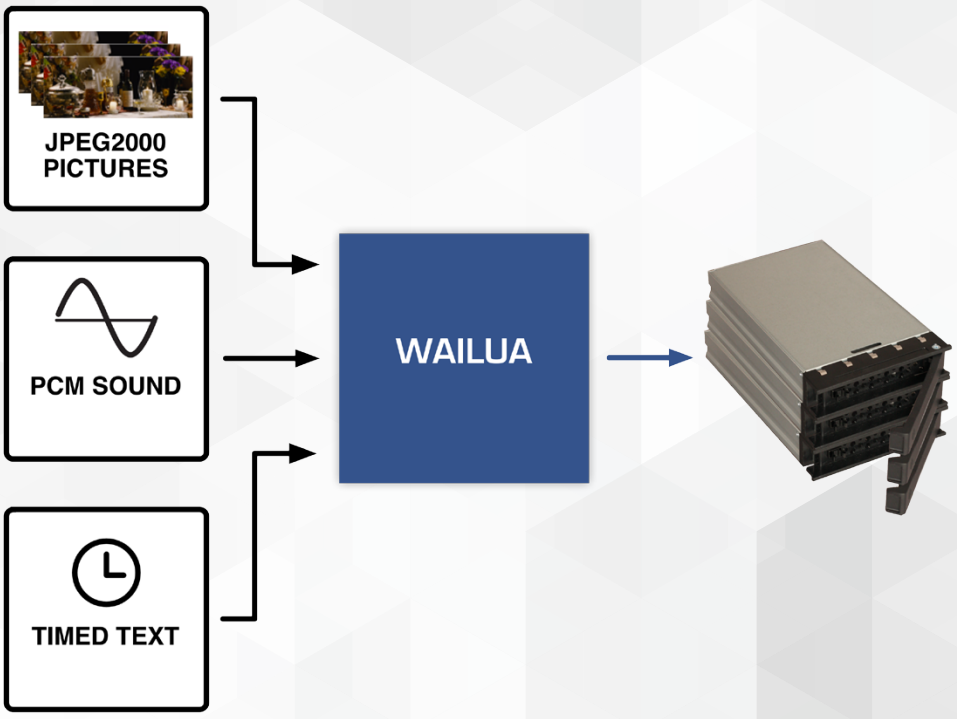
WAILUA

CineCert

D-Cinema Packaging Tools

WRAP ▪ ENCRYPT ▪ VALIDATE

- *Industry Standard Implementation*
- *Wrap JPEG2000, WAV, XML, PNG, TTF*
- *Flexible Encryption and KDM Features*
- *Test and Repair DCPs*
- *Easily Programmable and Scriptable*
- *Robust Python and C++ APIs*



CineCert's Wailua D-Cinema Mastering System is a package of software tools that allow the user to manipulate and test digital cinema files as part of a d-cinema packaging or distribution workflow. With the Wailua toolset you can easily create a D-Cinema Package (DCP) from source files by issuing individual commands or by scripting in your favorite shell, Python or C++ environment.

Wailua runs on your hardware and choice of operating system. Wailua is supported on over thirty variants of Linux and other common Unix-like platforms.

CineCert, LLC 2840 North Lima Street, Suite 110A, Burbank, CA 91504 USA
PHONE (818) 563-1455 **FAX** (818) 563-1459 **E-MAIL** sales@cinecert.com **WEB** www.cinecert.com

RELIABLE DIGITAL CINEMA

The Wailua D-Cinema Mastering System is a mature, enterprise-class DCP mastering solution. Top-tier distribution and post-production providers rely every day on Wailua for precise operations on D-Cinema Packages. Detailed, personal support is provided by our knowledgable technical staff.

Comprehensive

Wailua functions include creating/wrapping track files, making composition playlists (CPL) and packing lists, mapping a DCP to a storage volume, track file encryption, key management (KDM), file validation and analysis, certificate operations and many other features. Advanced tools are also provided to automate conform operations and supplemental package authoring.

Total Control

The Wailua tools are all command line operated so that they may be easily scripted in a Unix-like execution environment. Complete system functionality is also available to user-developed programs using the Wailua SDK for Python or C++. As shown in the examples to the right, the SDK enhances Wailua's extensive functionality by allowing you direct access to d-cinema objects and functions from an object-oriented programming environment. Full documentation of API and command line function is provided.

True File-Based Workflow

Wailua is ideal for implementing Service Oriented Architectures, batch processing, multiprocessing and other advanced workflow methodologies now common in post-production and mastering operations. Automated object creation, asset management integration and precisely customized metadata are easily supported. Wailua handles the details of d-cinema file processing, allowing you to concentrate on your application.

```python
#!/usr/bin/env python
import Wailua
#-- read CPL and signer info
with open("my-1000th-trailer.cpl.xml") as handle:
    xml_doc = handle.read()
content_authenticator = Wailua.X509Thumbprint()
cpl = Wailua.CompositionPlaylist()
cpl.decode_xml(xml_doc)
if cpl.SignatureMode != Wailua.SM_NONE:
    signer_chain, signer_thumbprint, subject_serial_number = \
                  Wailua.extract_dcp_signature_info(xml_doc)
if signer_chain:
    content_authenticator = signer_chain.list[0].Thumbprint

#-- load source keys
kdm = Wailua.KeyDeliveryMessage()
for filename in ("reel-1-picture.kdm.xml", "reel-2-picture.kdm.xml",
                 "reel-1-sound.kdm.xml", "reel-2-sound.kdm.xml"):
    with open(filename) as handle:
       kdm.decode_xml_for_target(handle.read())

#-- create consolidated KDM
kdm.ID.gen_random_value()
kdm.CompositionPlaylistID = cpl.ID
kdm.ContentTitle = cpl.ContentTitle
kdm.NotValidBefore.add_hours(-12)
kdm.NotValidAfter.decode_string(48)
kdm.DeviceList = [
    Wailua.X509Thumbprint("2jmj7l5rSw0yVb/vlWAYkK/YBwk=") ]
kdm.setup_with_cpl(cpl, content_authenticator)
print kdm.encode_xml_for_self()
```

```python
#!/usr/bin/env python
import Wailua
#-- make a Packing List
pkl = Wailua.PackingList()
pkl.ID.gen_random_value()
pkl.Creator = "AkbarJeff DCP 1.0 (now with Wailua v%s!)"
pkl.Creator += Wailua.version()
pkl.Issuer = options.issuer
pkl.Annotation = u"Not really meant for public consumption"
pkl.setup_with_file_list("my-1000th-trailer.cpl.xml") # recursive!
print pkl.encode_xml()
```

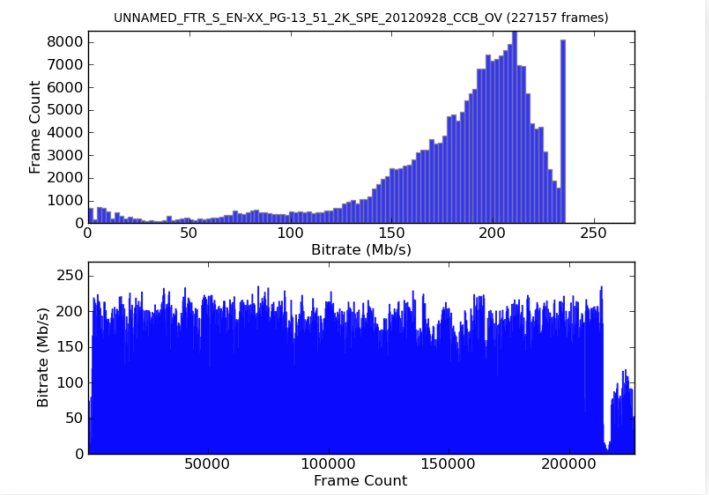
```python
#!/usr/bin/env python
import Wailua
#-- create RSA key and X.509 CSR
new_pk = Wailua.RSAPrivateKey()
new_pk.create_key_pair();
print new_pk.encode_pem()
csr = Wailua.X509CertificateRequest()
csr.set_public_key(new_pk)
csr.SubjectCommonName = "SM.coffeemaker-6000-sn-00838756"
csr.finalize()
print csr.encode_pem()
```

```python
#-- test a DCP volume
#!/usr/bin/env python
import sys
import Wailua
test_options = Wailua.TestOptions()
test_options.set_option("all-crypt")
test_options.set_option("no-xml-signature")
Wailua.test_file_list(sys.argv[1:], test_options, Wailua.DFT_Interop)
```

Ki'i

Ki'i, an optional Image Processing Module, can also be integrated with Wailua. It provides access to uncompressed image files, Color Transformation Language (CTL) support and JPEG2000 compression using a variety of software and hardware compression engines.

Ki'i allows the user to read and write Digital Source Master (DSM) images in TIFF, DPX, Cineon and OpenEXR formats, and to perform operations on those images such as color space and gamma transformations, compression, padding and cropping. Ki'i is packaged as a Python extension and a set of Python programs that perform common DSM processing tasks such as transforming DPX RGB images into JPEG2000 for d-cinema.



Bitrate Analysis