



高速部分画像検索キット
(FPGA アクセラレーション)

ユーザーマニュアル for Intel® Programmable
Acceleration Card with Intel® Arria® 10 GX FPGA

CA92344-2794-02

2019年12月

目次

1. 本書について.....	1
1.1. 本書の目的.....	1
1.2. 本書の読者.....	1
2. システム構成.....	2
2.1. 全体構成図.....	2
2.2. 動作環境及び必要なソフトウェア.....	3
3. FPGA アクセラレーションカードの設定.....	4
3.1. Intel® FPGA Program への登録.....	4
3.2. インテル(R) アクセラレーション・スタック (インテル(R) Xeon(R) CPU & FPGA 対応)のダウンロード.....	4
3.3. インテル(R) アクセラレーション・スタック (インテル(R) Xeon(R) CPU & FPGA 対応)のインストール.....	4
3.4. OPAE ライブラリのインストール.....	5
3.5. 'pac_a10'の Board Support Package の展開およびインストール.....	7
3.6. OpenCL ライブラリのインストール.....	7
4. 高速部分画像検索キットのインストール.....	7
4.1. 必要なソフトウェアのインストールと確認.....	7
4.2. 検索エンジンのインストール.....	10
4.3. tomcat への servlet 配備.....	10
4.4. tomcat と httpd の連携.....	11
4.5. サービスを有効にする.....	12
4.6. サービスの起動・停止・ステータス確認.....	12
5. FPGA アクセラレーションカードのハードウェア自己診断.....	13
5.1. FPGA アクセラレーションカードの搭載確認.....	13
5.2. FPGA Diagnostics Tool.....	13
5.2.1. fpgabist コマンドの実行.....	13
5.2.2. 温度の確認.....	15
5.2.3. エラーの確認.....	15
6. 設定ファイル.....	17
6.1. フォーマット.....	17
6.1.1. servlet プロパティファイル.....	17
6.1.2. 類似度計算パラメタファイル.....	18
6.1.3. servlet メッセージプロパティファイル.....	19
6.1.4. 画像検索実行結果ファイル.....	19
6.1.5. servlet のエラー記録ファイル.....	20
6.1.6. servlet ログファイル.....	20
6.1.7. 画像検索エンジンログファイル.....	21
7. 機能概要.....	22
7.1. 機能一覧.....	22
7.2. DB 画像 ID 一覧取得.....	23

7.2.1.	リクエストデータの形式.....	23
7.2.2.	ステータスコード.....	23
7.2.3.	レスポンスデータの形式.....	23
7.3.	DB 画像登録.....	24
7.3.1.	リクエストデータの形式.....	24
7.3.2.	ステータスコード.....	24
7.3.3.	レスポンスデータの形式.....	24
7.4.	DB 画像更新.....	25
7.4.1.	リクエストデータの形式.....	25
7.4.2.	ステータスコード.....	25
7.4.3.	レスポンスデータの形式.....	25
7.5.	DB 画像削除.....	26
7.5.1.	リクエストデータの形式.....	26
7.5.2.	ステータスコード.....	26
7.5.3.	レスポンスデータの形式.....	26
7.6.	画像検索.....	27
7.6.1.	リクエストデータの形式.....	27
7.6.2.	ステータスコード.....	28
7.6.3.	レスポンスデータの形式.....	28
7.7.	画像検索結果一覧取得.....	29
7.7.1.	リクエストデータの形式.....	29
7.7.2.	ステータスコード.....	29
7.7.3.	レスポンスデータの形式.....	29
7.8.	画像検索結果取得.....	30
7.8.1.	リクエストデータの形式.....	30
7.8.2.	ステータスコード.....	30
7.8.3.	レスポンスデータの形式.....	30
7.9.	画像検索結果削除.....	32
7.9.1.	リクエストデータの形式.....	32
7.9.2.	ステータスコード.....	32
7.9.3.	レスポンスデータの形式.....	32
8.	リソース.....	33
8.1.	エラーメッセージ.....	33
8.1.1.	凡例.....	33
8.1.2.	エラーメッセージ一覧.....	34
8.2.	定量制限.....	36

1. 本書について

1.1. 本書の目的

本書では高速で画像の検索を行う「高速部分画像検索キット(FPGA アクセラレーション)」の導入手順及び機能全般を説明します。

1.2. 本書の読者

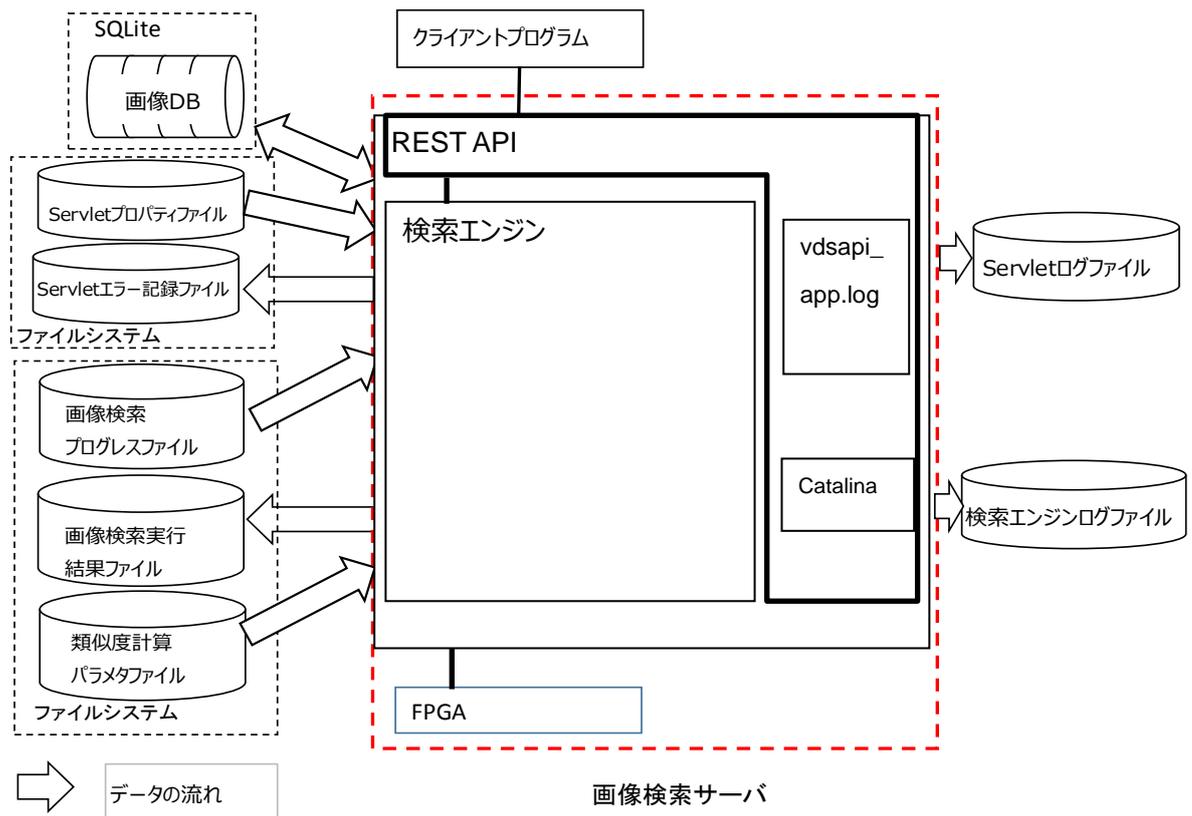
このマニュアルは、ハードウェアとソフトウェアについて十分な知識を持っているシステム管理者及びアプリケーションを開発するソフトウェア技術者を対象とします。

2. システム構成

2.1. 全体構成図

高速部分画像検索キットを搭載する画像検索サーバの全体構成を以下に示します。

図 2-1 画像検索サーバの全体構成



2.2. 動作環境及び必要なソフトウェア

検索エンジンを正しく動作させるために、以下のソフトウェアが必要です。

表 2-1 画像検索サーバの動作環境

項目	仕様
OS	Red Hat Enterprise Linux 7.6 (for Intel64)
データベース	SQLite3
HTTP サーバ	httpd, apache-tomcat 8.5.46
Java	OpenJDK 8 (1.8)
画像処理ライブラリ	OpenCV 3.2
Intel® Acceleration Stack	Acceleration Stack 1.2

本書では、Red Hat Enterprise Linux 7.6 のパッケージとして、以下をインストールすることを前提に手順を記載しています。

- ベース環境
 - サーバー（GUI 使用）
- 選択した環境のアドオン
 - ハードウェアモニタリングユーティリティ
 - パフォーマンスツール
 - 互換性ライブラリ
 - 開発ツール

3. FPGA アクセラレーションカードの設定

インテル社のダウンロードページより、必要なモジュールのダウンロードおよびインストールを行います。

※ご購入時は工場にてインストール済みのため、インストール作業は不要です。

3.1. Intel® FPGA Program への登録

Intel®製品をダウンロードするには Intel® FPGA Program への登録が必要です。Intel® Customer または Partner アカウントにサインイン済の場合は追加登録することも可能です。

<https://www.intel.com/content/www/us/en/my-intel/fpga-sign-in.html>

3.2. インテル(R) アクセラレーション・スタック (インテル(R) Xeon(R) CPU & FPGA 対応)のダウンロード

下記 URL より、Acceleration Stack 1.2 for Runtime をダウンロードしてください。ダウンロードには Intel® FPGA Program へのサインインが必要です。

<https://www.intel.com/content/www/us/en/programmable/solutions/acceleration-hub/downloads.html>

本製品は Acceleration Stack 1.2 にて動作確認を実施しています。上記ダウンロードページにて異なるバージョンのソフトウェアが配布されている場合は、下記アーカイブページより Acceleration Stack 1.2 をダウンロードしてください。

<https://www.intel.com/content/www/us/en/programmable/solutions/acceleration-hub/archives.html>

3.3. インテル(R) アクセラレーション・スタック (インテル(R) Xeon(R) CPU & FPGA 対応)のインストール

1. ダウンロードした Acceleration Stack for Runtime を任意のフォルダにコピーして展開します。

```
$ tar xvf a10_gx_pac_ias_1_2_pv_rte_installer.tar.gz
$ cd a10_gx_pac_ias_1_2_pv_rte_installer
```

2. セットアップシェルを実行します。

```
$ ./setup.sh
```

3. “Do you want to continue to install the software?” 表示に対して [y]を入力します。
4. “Do you wish to install the OPAAE?”表示に対して[n]を入力します。
 - a. OPAAE のインストールは以降の手順にて実施します
5. プログラム使用に関するライセンスの確認メッセージが表示されます。ライセンス内容を確認し、「Accept」を行ってください。

6. プログラムをインストールするディレクトリの確認が表示されます。

初期設定ディレクトリ: “/home/<username>/intelrtestack”にインストールすることを前提に説明します。

7. プログラム動作環境の設定を行います。

```
$ source /home/<username>/intelrtestack/init_env.sh
```

3.4. OPAE ライブラリのインストール

8. 以下のパッケージをインストールします。

```
$ sudo yum install gcc gcc-c++ cmake make autoconf automake libxml2¥  
libxml2-devel json-c-devel boost ncurses ncurses-devel¥  
ncurses-libs boost-devel libuuid libuuid-devel python2-jsonschema¥  
doxygen hwloc-devel libpng12 rsync python2-pip
```

※"json-c-devel", "python2-jsonschema", "python-repoze-lru", "hwloc-devel", "python2-pip"および"intelhex"は extra パッケージのため、オンライン環境、または個別ダウンロードが必要です。

```
$ sudo yum install json-c-devel-*.x86_64.rpm python2-jsonschema-*.rpm¥  
python-repoze-lru-*.rpm hwloc-devel-*.rpm python2-pip-*.rpm (extra パッケージ)
```

```
$ sudo pip install intelhex
```

9. FPGA ドライバをインストールします。

- a. 旧版の OPAE ライブラリが存在する場合、削除します。

```
sudo yum remove opae*.x86_64
```

- b. Extra Packages for Enterprise Linux (EPEL)をインストールします。

```
$ sudo yum install epel-release
```

※"epel-release"がインストールできない場合、以下をインストールしてから、opae ソフトウェアをインストールします。

dkms:2.2.0.2 以降(下記は 2.6.1-1 を使用した例)

```
$ sudo yum install elfutils-libelf-devel (OS package)  
$ sudo yum install dkms-2.6.1-1.el7.noarch.rpm (extra)
```

- c. OPAE ソフトウェアのインストールディレクトリに移動します。

```
cd $OPAE_PLATFORM_ROOT/sw
```

- d. ドライバをインストールします。

```
sudo yum install opae-intel-fpga*.rpm
```

10. OPAE ライブラリをインストールします。

```
$ cd $OPAE_PLATFORM_ROOT/sw/  
$ sudo yum install opae*.rpm
```

11. fpgaflash を/usr/bin にコピーします。

```
sudo cp fpgaflash /usr/bin
sudo chmod 755 /usr/bin/fpgaflash
```

12. ライブラリパスを更新します。

```
$ sudo ldconfig
```

13. Linux kernel のインストールを確認します。

```
lsmod |grep fpga
```

表示例は下記となります。

```
intel_fpga_pac_iopll      13392  0
intel_fpga_pac_hssi      18347  0
intel_fpga_fme           54120  0
intel_fpga_afu           32062  0
intel_fpga_pci           26439  2 intel_fpga_afu,intel_fpga_fme
fpga_mgr_mod             14693  1 intel_fpga_fme
```

14. インストールされた rpm を確認します。

```
rpm -qa |grep opae
```

表示例は下記となります。

```
opae-tools-1.1.2-1.x86_64
opae-devel-1.1.2-1.x86_64
opae-libs-1.1.2-1.x86_64
opae-1.1.2-1.x86_64
opae-tools-extra-1.1.2-1.x86_64
opae-intel-fpga-driver-1.1.2-1.x86_64
opae-ase-1.1.2-1.x86_64
```

15. FPGA Interface Manager (FIM)のバージョンは下記となります。

```
$ fpgainfo fme
Board Management Controller, microcontroller FW version 26889
Last Power Down Cause: POK_CORE
Last Reset Cause: External reset
Power-on-reset
//***** FME *****/
Object Id                : 0xEF00000
PCIe s:b:d:f             : 0000:18:00:0
Device Id                : 0x09C4
Socket Id                : 0x00
Ports Num                : 01
Bitstream Id             : 0x123000200000185
```

Bitstream Version	: 0x55C900030201
Pr Interface Id	: 69528db6-eb31-577a-8c36-68f9faa081f6

3.5. 'pac_a10'の Board Support Package の展開およびインストール

1. FPGA アクセラレーションカードのサポートパッケージを展開します。

```
$ cd $OPAE_PLATFORM_ROOT/openccl
$ tar xf openccl_bsp*.tar.gz
$ cd openccl_bsp
$ export AOCL_BOARD_PACKAGE_ROOT=`pwd`
```

2. サポートパッケージをインストールします。

```
$ sudo mkdir -p /opt/Intel/OpenCL/Boards
$ sudo chown -R fpga:fpga /opt/Intel
$ aocl install
```

3.6. OpenCL ライブラリのインストール

1. OpenCL ライブラリをインストールします。

```
$ sudo mkdir -p /etc/OpenCL/vendors
$ sudo cp /home/fpga/intelrtestack/openccl_rte/aclrte-linux64/Altera.icd /etc/OpenCL/vendors
```

4. 高速部分画像検索キットのインストール

高速部分画像検索キットのインストール手順を以下に示します。

※ご購入時は工場にてインストール済みのため、インストール作業は不要です。

注意：高速部分画像検索キットを動作させるためには、一般ユーザが必要になります。ここでは ユーザ名/グループ: fpga/fpga を例として記載します。

4.1. 必要なソフトウェアのインストールと確認

1. 以下のコマンドでパッケージがインストール済みか確認し、インストールされていなかった場合は、追加でインストールしてください

Sqlite

```
$ yum list installed | grep sqlite
```

OpenJDK

```
$ yum list installed | grep java
```

Httpd

```
$ yum list installed | grep httpd
```

- 以下のソフトウェアのインストールを実施します。
この時、インターネット接続環境であることが必要になります。

OpenCV

以下の URL に従い、OpenCV のインストールを実施します。

http://docs.opencv.org/3.0-last-rst/doc/tutorials/introduction/linux_install/linux_install.html

- 共有ライブラリの登録

/etc/ld.so.conf.d にファイルを作成します。

ファイル名 : opencv.conf

内容

```
/opt/opencv/lib
```

- ライブラリパスを更新します。

```
$ sudo ldconfig
```

- Tomcat の設定

以下のコマンドで Tomcat をインストールします。

```
$ cd /opt
$ sudo mkdir apache-tomcat
$ sudo chown [fpga]:[fpga] apache-tomcat
$ cp apache-tomcat-8.5.46.tar.gz /opt/apache-tomcat
$ cd apache-tomcat
$ tar zxvf apache-tomcat-8.5.46.tar.gz
```

以下のパスに Unit ファイルを作成します。赤字の箇所は環境に合わせて編集します。

```
/etc/systemd/system/tomcat8.service
```

```
[Unit]
Description=Apache Tomcat 8
After=network.target

[Service]
User=fpga
Group=fpga
Type=oneshot
PIDFile=/opt/apache-tomcat/apache-tomcat-8.5.46/tomcat.pid
RemainAfterExit=yes
WorkingDirectory=/home/fpga
LimitMEMLOCK=infinity

ExecStart=/opt/apache-tomcat/apache-tomcat-8.5.46/bin/startup.sh
```

```
ExecStop=/opt/apache-tomcat/apache-tomcat-8.5.46/bin/shutdown.sh 20
```

```
[Install]
```

```
WantedBy=multi-user.target
```

6. 作成したファイルの権限を変更します。

```
$ sudo chmod 664 /etc/systemd/system/tomcat8.service
```

7. tomcat 起動時のメモリ割り当ての設定

次のファイルの編集を行うこと。黄色箇所が追加対象になります。

```
/opt/apache-tomcat/apache-tomcat-8.5.46/bin/catalina.sh
```

割り当てるメモリ容量については、環境に合わせて変更ください。

```
#!/bin/sh
```

```
(略)
```

```
# CATALINA_OPTS (Optional) Java runtime options used when the "start",  
# "run" or "debug" command is executed.
```

```
# Include here and not in JAVA_OPTS all options, that should  
# only be used by Tomcat itself, not by the stop process,  
# the version command etc.
```

```
# Examples are heap size, GC logging, JMX ports etc.
```

```
#
```

```
CATALINA_OPTS="-server -Xms8192m -Xmx8192m -XX:PermSize=128m -XX:MaxPermSize=256m  
-XX:MaxNewSize=2048m -Dorg.apache.catalina.connector.CoyoteAdapter.ALLOW_BACKSLASH=true"
```

```
#
```

```
# CATALINA_TMPDIR (Optional) Directory path location of temporary directory
```

```
(略)
```

```
#
```

```
# JAVA_OPTS (Optional) Java runtime options used when any command  
# is executed.
```

```
# Include here and not in CATALINA_OPTS all options, that  
# should be used by Tomcat and also by the stop process,  
# the version command etc.
```

```
# Most options should go into CATALINA_OPTS.
```

```
#
```

```
JAVA_OPTS="-Dfile.encoding=UTF-8"
```

```
#
```

```
# JAVA_ENDORSED_DIRS (Optional) Lists of of colon separated directories
```

```
(略)
```

```
#
```

```
# CATALINA_PID (Optional) Path of the file which should contains the pid
```

```
#           of the catalina startup java process, when start (fork) is
#           used
CATALINA_PID="/opt/apache-tomcat/apache-tomcat-8.5.46/tomcat8.pid"
# LOGGING_CONFIG (Optional) Override Tomcat's logging config file
#           Example (all one line)
```

4.2. 検索エンジンのインストール

1. 添付 CD の 'pir_kit_V03R10' ディレクトリを任意のディレクトリにコピーします。

```
$ cp -r /mnt/cdrom/pir_kit_V03R10/ .
```

2. 検索エンジンをインストールします。

```
$ cd pir_kit_V03R10
$ sudo rpm -ivh --nodeps pir_servlet-2.0.3-1.x86_64.rpm
```

3. インストールした検索エンジンの権限を変更します。ユーザが fpga:fpga の場合、以下のコマンドを実行します。

```
$ sudo chown -R fpga:fpga /opt/FJpir
```

参考 : pir_kit のバージョン確認コマンド

```
$ yum list | grep pir_servlet
```

4. bw2tk パッケージの前提パッケージをインストールします。

```
$ sudo yum install qt-x11
$ sudo yum install libusb
```

5. bw2tk パッケージをインストールします。

```
$ sudo yum install bw2tk-lite-2018.6.el7.x86_64.rpm
```

参考 : bw2tk パッケージのバージョン確認コマンド

```
$ yum list | grep bw2tk
```

4.3. tomcat への servlet 配備

1. Tomcat に検索エンジンの機能を追加します。

```
$ mkdir /opt/apache-tomcat/apache-tomcat-8.5.46/conf/Catalina
$ mkdir /opt/apache-tomcat/apache-tomcat-8.5.46/conf/Catalina/localhost/
$ cp /opt/FJpir/etc/vdsapi.xml /opt/apache-tomcat/apache-tomcat-8.5.46/conf/Catalina/localhost/
```

4.4. tomcat と httpd の連携

1. /etc/httpd/conf/httpd.conf を編集します。
 - a. 内容：ファイル末尾に以下の行を追加します。

```
<Location /vdsapi/>
    ProxyPass ajp://localhost:8009/vdsapi/ nocanon
    Header set Access-Control-Allow-Origin "*"
    Header set Access-Control-Allow-Methods "POST, GET, DELETE, PUT, OPTIONS"
    Header set Access-Control-Allow-Headers "X-PINGOTHER, Content-Type"
</Location>
```

- b. /etc/httpd/conf.modules.d/配下のファイルを確認し、記載がない場合は下記部分に追記します。

※環境によっては/etc/httpd/conf.modules.d/配下のファイル名の記載がある場合があります。

この場合は“proxy_ajp_module modules/mod_proxy_ajp.so”を追記します。

編集例:

```
LoadModule proxy_ajp_module modules/mod_proxy_ajp.so
```

2. Tomcat サービスの起動時に必要な初期化処理を追加します。

"/opt/apache-tomcat/apache-tomcat-8.5.46/bin/startup.sh"に以下を追記します。

```
#!/bin/sh
(略)
# -----
# Start Script for the CATALINA Server
# -----
# for pir_servlet openc1 version
source /home/fpga/intelrtestack/init_env.sh
source /home/fpga/intelrtestack/a10_gx_pac_ias_1_2_pv/openc1/openc1_bsp/linux64/libexec/setu
p_permissions.sh
aocl program aocl0 /opt/FJpir/device/match.aocx
export CL_CONTEXT_COMPILER_MODE_INTELFPGA=3
```

3. サービス起動時に管理者権限の実行処理を行うため、ユーザ“fpga”が、パスワード無しで、“sudo”実行を行えるようにします。以下のコマンドで、sudoers ファイルを編集します。

```
$ sudo visudo
```

4. パスワード無しで、“sudo”実行ができるユーザとして“fpga”を追加します。

```
## Same thing without a password
# %wheel          ALL=(ALL)          NOPASSWD: ALL
fpga             ALL=(ALL)          NOPASSWD: ALL
```

4.5. サービスを有効にする

1. Apache(httpd)サービスの自動起動設定を行います。

```
$ sudo systemctl enable httpd.service  
$ sudo systemctl start httpd.service  
$ sudo systemctl enable tomcat8.service
```

4.6. サービスの起動・停止・ステータス確認

サービスの起動

```
$ sudo systemctl start tomcat8.service
```

サービスの停止

```
$ sudo systemctl stop tomcat8
```

サービスのステータス確認

```
$ sudo systemctl status tomcat8
```

5. FPGA アクセラレーションカードのハードウェア自己診断

搭載されている FPGA アクセラレーションカードのハードウェアに不具合が考えられる場合、本項の手順にてハードウェアの自己診断を行うことができます。

5.1. FPGA アクセラレーションカードの搭載確認

FPGA アクセラレーションカードがオペレーティングシステムから正しく認識されているかを確認します。認識されていない場合は、何も出力されません。この場合、PRIMERGY の故障・修理の相談窓口にお問い合わせください。

```
$ lspci |grep 09c4
```

正常な出力例：

```
18:00.0 Processing accelerators: Intel Corporation Device 09c4
```

5.2. FPGA Diagnostics Tool

fpgabist コマンドにより、FPGA アクセラレーションカードのハードウェア自己診断を実行することができます。

fpgabist コマンドは、Intel® Acceleration Stack に含まれています。

5.2.1. fpgabist コマンドの実行

fpgabist コマンドを使用して、自己診断を実行してください。各診断は数十秒で完了します。

※ fpgabist による自己診断は、本製品の FPGA 部へのデータ書き込みを行うため、実行中の FPGA アプリケーションを停止してから実行してください。

※ fpgabist の実行時に「Could not allocate buffer」等メモリ不足のエラーが出力される場合は、下記コマンドにて 20 ページのメモリを 2MB 割り当ててください。

```
$ sudo sh -c "echo 20 > /sys/kernel/mm/hugepages/hugepages-2048kB/nr_hugepages"
```

※ fpgabist の実行時に「**Exception caught: stoi - could not convert af to a number**」「**Error: device enumeration failed. Please make sure that the driver is loaded and that a bitstream for AFU id:**

xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx is programmed.」といったエラーが出力される場合は、下記手順にて問題を修正してください。

- a. ファイル bist_nlb3.py を編集します。

```
$ sudo vim /usr/bin/bist_nlb3.py
```

- b. 48 行目を編集します。

```
cmd = "fpgadiag -B {} {}".format(bus_num, param)
---
cmd = "fpgadiag -B 0x{} {}".format(bus_num, param)
```

1. 下記コマンドにて host memory interface の診断を実行します。

```
sudo fpgabist $OPAE_PLATFORM_ROOT/hw/samples/nlb_mode_3/bin/nlb_mode_3.gbs
```

末尾が下記出力例のように「Built-in Self-Test Completed.」となっている場合、host memory interface の診断は完了しています。

```
Cachelines Read_Count Write_Count Cache_Rd_Hit Cache_Wr_Hit Cache_Rd_Miss
Cache_Wr_Miss Eviction 'Clocks(@400 MHz)' Rd_Bandwidth Wr_Bandwidth
1024 480797340 488815296 0 0
0 0 0 1000021563 6.234 GB/s 6.256 GB/s
VH0_Rd_Count VH0_Wr_Count VH1_Rd_Count VH1_Wr_Count VL0_Rd_Count
VL0_Wr_Count480797340 488815297 0
0 0 0
Built-in Self-Test Completed.
```

2. 下記コマンドにて local memory interface の診断を実行します。

```
sudo fpgabist $OPAE_PLATFORM_ROOT/hw/samples/dma_afu/bin/dma_afu.gbs
```

末尾が下記出力例のように「Finished Executing DMA Tests」となっている場合、local memory interface の診断は完了しています。

```
Running test in HW mode
Buffer Verification Success!
Buffer Verification Success!
Running DDR sweep test
Allocated test buffer
Fill test buffer
DDR Sweep Host to FPGA
Measured bandwidth = 6616.881910 Megabytes/sec
Clear buffer
DDR Sweep FPGA to Host
Measured bandwidth = 6932.201347 Megabytes/sec
Verifying buffer.
Buffer Verification Success!
Finished Executing DMA Tests
```

5.2.2. 温度の確認

fpgabist の実行結果の THERMAL セクションの Temperature 項より、チップ温度を確認してください。

```
//***** THERMAL *****/
Class Path      : /sys/class/fpga/intel-fpga-dev.0/intel-fpga-fme.0/thermal_mgmt
Device Path     :
                /sys/devices/pci0000:ae/0000:ae:00.0/0000:af:00.0/fpga/intel-fpga-dev.0/intel-fpga-
                fme.0/thermal_mgmt
Bus            : 0xAF
Device         : 0x00
Function       : 0x00
Device Id      : 0x09C4
Temperature    : 43° C
```

※高速部分画像検索の実行中に温度の確認を行いたい場合は、下記コマンドを実行してください。fpgabist の THERMAL セクションと同一の結果が出力されます。

```
$ sudo fpgainfo temp
```

Operating Temperature は 95°Cまで、100°Cに達すると FPGA アクセラレーションカードはシャットダウンされます。本項目によるハードウェア自己診断にて 95°Cを超える温度が検出された場合、環境温度やシステムファン等の関連コンポーネントの状態を確認の上、PRIMERGY の故障・修理の相談窓口にお問い合わせください。

<http://www.fujitsu.com/jp/support/bussiness/repaircenter/primergy/>

5.2.3. エラーの確認

fpgabist のその他実行結果に文字列「Error」、「Failed test」、「[ERROR][nlb3]」または「Invalid data at」が含まれている場合、エラーが検出されています。

ただし、下記例のように、Errors 表記の行に 0x0 と表示されている場合は、そのエラーがないことを示しています。

```
//***** FME ERRORS *****/
Object Id      : 0xED00000
PCIe s:b:d:f   : 0000:17:00:0
Device Id      : 0x09C4
Socket Id      : 0x00
Ports Num      : 01
Bitstream Id   : 0x123000200000185
Bitstream Version : 0x7FFD00030201
Pr Interface Id : 69528db6-eb31-577a-8c36-68f9faa081f6
First Error    : 0x0
Next Error     : 0x0
Errors         : 0x0
PCIe1 Errors   : 0x0
Nonfatal Errors : 0x0
```

PCIe error detected	
Inject Error	: 0x0
Catfatal Errors	: 0x0
PCIe0 Errors	: 0x0

※ 下記例のように、FME ERRORS セクションにて 0x1 の First Error, 0x4 の Nonfatal Errors および 0x1 の PCIe Errors が検出される場合がありますが、その他項目の Error カウント数が 0x0 表示となっていれば、本エラーはハードウェア故障を示すものではなく、問題ありません。

//***** FME ERRORS *****//	
Object Id	: 0xEF00000
PCIe s:b:d:f	: 0000:18:00:0
Device Id	: 0x09C4
Socket Id	: 0x00
Ports Num	: 01
Bitstream Id	: 0x123000200000185
Bitstream Version	: 0x7FFF00030201
Pr Interface Id	: 69528db6-eb31-577a-8c36-68f9faa081f6
First Error	: 0x1
Next Error	: 0x0
Errors	: 0x0
PCIe1 Errors	: 0x0
Nonfatal Errors	: 0x4
PCIe error detected	
Inject Error	: 0x0
Catfatal Errors	: 0x0
PCIe0 Errors	: 0x1
TLP format/type error detected	

本項目によるハードウェア自己診断にてエラーが検出された場合は、PRIMERGY の故障・修理の相談窓口にお問い合わせください。

<http://www.fujitsu.com/jp/support/bussiness/repaircenter/primergy/>

6. 設定ファイル

6.1. フォーマット

6.1.1. servlet プロパティファイル

部分画像検索 servlet の実行環境を定義した「key=value」形式のファイル。

6.1.1.1. 設定キー一覧

指定可能なプロパティを下表に示します。

表 6-1 プロパティファイルのキー一覧

キー名	指定値意味・指定方法	デフォルト値
response.charset	レスポンスの Content-Type に設定する文字セット	utf-8
search.target_image.filename	画像検索リクエストで、クライアントから送られた画像ストリームデータから画像ファイルを復元する際のファイル名	fpgaTargetImage.png
search.model.unit	フレームの分割数 検索単位の最大数(1～1000000)。	10000
search.request.queue.capacity	検索リクエストキューの大きさ(1～1000000)。	50
search.exec.queue.capacity	検索実行キューの大きさ(1～1000)。	3
image.id.max.size.byte	画像 ID の最大サイズ(byte)(1～8388608)。	8388608 (8MB)
image.file.max.size.byte	画像ファイルの最大サイズ(byte)(1～1073741824)。	1073741824(1GB)
model_images.dir.path	クライアントから送られた画像ストリームデータから画像ファイルを復元する際のディレクトリパス名の末尾は、/で終わる必要がある	
result.dir.path	画像検索の実行結果を格納するディレクトリパス名 パス名の末尾は、/で終わる必要がある	/opt/Fjpir/result/
result.status.filename	画像検索プログレスファイル名	status.txt
result.result.filename	画像検索実行結果ファイル名	results.txt
result.error.filename	servlet のエラー記録ファイル名	error.txt
ini.file.path	類似度計算パラメタファイル名	

キー名	指定値意味・指定方法	デフォルト値
keypoint.lower.limit	イメージ画像から算出する特徴点個数の下限値(4~11284)。 ベタ画像(solid image)等の画像イメージに特徴がないと特徴点個数が小さい値となる。 この値を下回る場合、画像検索処理が簡易処理で行われる。	4
search.inmemory	インメモリモード(初期処理時に画像 DB の内容をメモリに展開するモード)の設定値 true:インメモリモード, false: 二次記憶モード	true
memory.base.size.byte	画像検索サーバが起動で使用する基本メモリ使用量。	105232384 (102,766KiB)
memory.1search.size.byte	検索 1 回に使用するメモリ使用量。	629252536 (614,504.43KiB)
memory.1db.size.byte	DB 画像 1 件で使用するメモリ使用量。	28088(27.43KiB)
search.timeout.second	タイムアウト(秒) (1~3600)	10(秒)
search.map.threshold	検索残回数閾値	10000(回)
search.map.base	Tomcat 起動で使用するメモリマップ数	1000
search.map.use	検索 1 回で使用するメモリマップ数	2

リスト 6-1 プロパティファイル サンプル

```

response.charset=utf-8
search.target_image.filename=fpgaTargetImage.png
search.model.unit=10000
search.request.queue.capacity=50
search.exec.queue.capacity=3
model_images.dir.path=/usr/home/FPGA/rest_pir_server/model_images/
result.dir.path=/usr/home/FPGA/rest_pir_server/results/
result.status.filename=status.txt
result.result.filename=results.txt
ini.file.path=/usr/home/FPGA/rest_pir_server/pir_server.ini
debug=false
search.inmemory=true

```

6.1.2. 類似度計算パラメタファイル

画像検索エンジンの実行環境を定義した「key=value」形式のファイル。検索エンジンにおいて、このファイルの指定値が許容された値であるか上限値および下限値をチェックします。

6.1.2.1. 設定キー一覧

指定可能なパラメータを下表に示します。

表 6-2 類似度計算パラメタファイルのキー一覧 上限値とエラーチェックする

キー名	指定値意味・指定方法	デフォルト値
initial_target_width	FPGA へ送るクエリのピラミッド画像 1 段の横幅条。上限 800、下限 100	800
resize_level	クエリ画像のピラミッド画像の階層数。上限 14、下限 1	14
coarse_detection_thresh	粗いマッチングに使用する閾値。上限 1.0、下限 0	0.07
fine_detection_thresh	細かいマッチングに使用する閾値。上限 1.0、下限 0	0.1
loglevel	検索エンジンログの出力レベル。指定値 0:ALL, 1:WARN, 2:ERROR	0

リスト 6-2 類似度計算パラメタファイル サンプル

```
coarse_detection_thresh = 0.07
fine_detection_thresh = 0.1
initial_target_width = 1
resize_level = 0
loglevel = 0
keypoint.lower.limit = 4
```

6.1.3. servlet メッセージプロパティファイル

本ファイルは、「key=value」形式で、エラーID とエラーメッセージを格納したファイルです。

リスト 6-3 servlet メッセージプロパティファイル サンプル

```
message.N0001=Your image was registered successfully.
(中略)
error_message.E0001=An internal server error occurred.
error_message.E0002=An illegal request was sent.
(中略)
error_message.E0018=Your image ID is too long.
error_message.E0019=Your image ID contains not allowed characters.
```

6.1.4. 画像検索実行結果ファイル

検索エンジン側の検索処理で、類似画像を検出した時の結果を格納します。

類似画像を検出しなかった場合は、このファイルは作成されません。

クライアントプログラムからの「画像検索結果取得」要求時に、検索結果を確認するため、servlet がこのファイルを読み込みます。

6.1.4.1. 画像検索実行結果ファイルのデータ

画像検索実行結果ファイルには、以下のデータを 1 レコードに 1 セットを格納します。

検出した類似画像が複数の存在する場合、複数レコードを出力します。

表 6-3 画像検索プログレスファイルのデータ一覧

キー名	型	内容
id	文字列	画像検索で検出した画像 ID
similarity	数値(小数付き)	類似度(1.0 を 100%とした 1.0 以下の値)
ttlx	数値(小数付き)	クエリ画像 マッチ矩形の x 座標の最小値
ttly	数値(小数付き)	クエリ画像 マッチ矩形の y 座標の最小値
tbrx	数値(小数付き)	クエリ画像 マッチ矩形の x 座標の最大値
tbry	数値(小数付き)	クエリ画像 マッチ矩形の y 座標の最大値
mtlx	数値(小数付き)	DB 画像 マッチ矩形の x 座標の最小値
mtly	数値(小数付き)	DB 画像 マッチ矩形の y 座標の最小値
mbrx	数値(小数付き)	DB 画像 マッチ矩形の x 座標の最大値
mbry	数値(小数付き)	DB 画像 マッチ矩形の y 座標の最大値

6.1.5. servlet のエラー記録ファイル

servlet において検出したエラーを格納します。

6.1.5.1. 画像検索実行結果ファイルのデータ

servlet のエラー記録ファイルには、以下のデータ 1 セットを格納します。

複数のエラーを検出した場合も、最初に発生したエラー内容のみを格納します。

表 6-4 servlet のエラー記録ファイルデータ一覧

キー名	型	内容
HttpErrorCode	数値	HTTP Status Code
ErrorMessageCode	文字列	エラーメッセージコード

6.1.6. servlet ログファイル

servlet の実行ログを格納します。

6.1.6.1. servlet ログファイルのデータ

servlet ログファイルは、log4j2.xml で指定されたログレベル以上のログを格納します。

リスト 6-4 servlet ログファイル サンプル

```

2017-03-23 13:54:46.011 [localhost-startStop-1] DEBUG
[com.fujitsu.vds.vdsapi.util.Initializer#contextInitialized] start initialize
2017-03-23 13:54:46.016 [localhost-startStop-1] TRACE
[com.fujitsu.vds.vdsapi.logic.task.SearchTaskManager#startUp] Start.
2017-03-23 13:54:46.016 [localhost-startStop-1] INFO
[com.fujitsu.vds.vdsapi.logic.task.SearchTaskManager#startUp] start up
SearchTask
    
```

```
2017-03-23 13:54:46.018 [localhost-startStop-1] TRACE
[com.fujitsu.vds.vdsapi.logic.task.SearchTaskManager#startup] End.
```

6.1.7. 画像検索エンジンログファイル

部分画像検索 servlet および画像検索エンジンが標準出力に出力した実行ログを格納します。

6.1.7.1. 画像検索エンジンログファイルのデータ

画像検索エンジンログファイルには、以下のデータを 1 レコードに 1 セットを格納します。

複数のエラーを検出した場合、複数レコードを出力します。

表 6-5 画像検索エンジンログファイルのデータ一覧

キー名	型	内容
HttpErrorCode	数値	HTTP Status Code
ErrorMessageCode	文字列	エラーID
ErrorMessage	文字列	エラーメッセージ

7. 機能概要

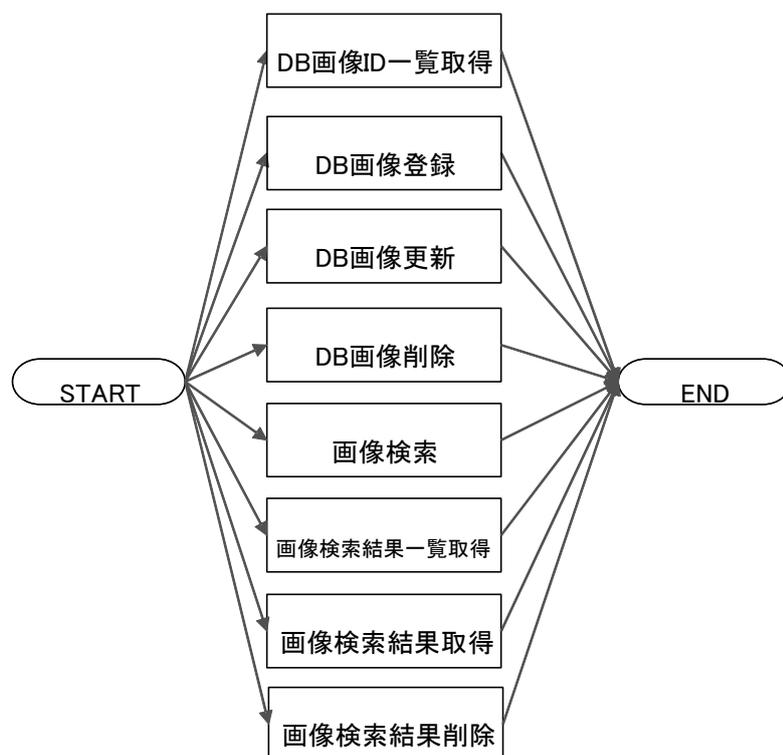
7.1. 機能一覧

画像検索サーバが提供する機能を下表に示します。

表 7-1 画像検索サーバの機能

アクション	呼出し形式 { }:URLパス []:埋め込み 《》:ファイルデータ []:繰り返し	メソッド
DB 画像 ID 一覧取得	http://example.com/vdsapi/images/	GET
DB 画像登録	http://example.com/vdsapi/images/ id=【画像 ID】 file=《画像ファイル》	POST
DB 画像更新	http://example.com/vdsapi/images/{画像 ID} file=《画像ファイル》	PUT
DB 画像削除	http://example.com/vdsapi/images/{画像 ID}	DELETE
画像検索	http://example.com /vdsapi/search/ query=《クエリ画像ファイル》 ids=【DB 画像 ID1】 [ids=【DB 画像 ID2】]...	POST
画像検索結果一覧取得	http://example.com/vdsapi/results/	GET
画像検索結果取得	http://example.com/vdsapi/results/{画像検索リクエスト ID}	GET
画像検索結果削除	http://example.com/vdsapi/results/{画像検索リクエスト ID}	DELETE

図 7-1 画像検索サーバの提供機能の検索エンジンの関係



以下、各機能について説明を記述します。

7.2. DB 画像 ID 一覧取得

画像検索サーバに登録済の画像 ID リストを通知します。

7.2.1. リクエストデータの形式

メソッドタイプ:GET

リクエスト URL: <http://example.com/vdsapi/images/>

表 7-2 DB 画像 ID 一覧取得のリクエストデータ

カラム名	説明	備考
-	-	

7.2.2. ステータスコード

表 7-3 DB 画像 ID 一覧取得のステータスコード

ステータスコード	説明	備考
200	画像 ID のリストを返却する	
500	取得失敗	サーバ運営者に対応を依頼

7.2.3. レスポンスデータの形式

表 7-4 画像 ID 一覧取得のレスポンスデータ

カラム名	説明	備考
searchlist	画像 ID の一覧	

7.3. DB 画像登録

画像データの特徴量を算出して、特徴量を登録します。

7.3.1. リクエストデータの形式

メソッドタイプ:POST

リクエスト URL: `http://example.com/vdsapi/images/ id=【画像 ID】 file=《画像ファイル》`

表 7-5 DB 画像登録のリクエストデータ

カラム名	説明	備考
id	画像 ID	
InputStream fileStream	画像データの入力ストリーム	

7.3.2. ステータスコード

表 7-6 DB 画像登録のステータスコード

ステータスコード	説明	備考
200	登録成功	
400	登録失敗. 不正なパラメータ	
409	登録失敗. 画像 ID が登録済み	
500	登録失敗. 検索エンジンの呼出しでエラーを返却	サーバ運営者に対応を依頼
503	登録失敗. 登録可能枚数を超過	

7.3.3. レスポンスデータの形式

表 7-7 DB 画像登録のレスポンスデータ

カラム名	説明	備考
-	-	-

7.4. DB 画像更新

画像更新は、登録されている画像 ID に対して、再度、画像データの特徴量を算出して、更新を行います。

7.4.1. リクエストデータの形式

メソッドタイプ:PUT

リクエスト URL: `http://example.com/vdsapi/images/{画像 ID} file=《画像ファイル》`

表 7-8 DB 画像更新のリクエストデータ

カラム名	説明	備考
id	画像 ID	
InputStream fileStream	画像データの入力ストリーム	

7.4.2. ステータスコード

表 7-9 DB 画像更新のステータスコード

ステータスコード	説明	備考
200	更新成功	
400	更新失敗 不正なパラメータ	
404	更新失敗 対象となる画像 ID が存在しない	
500	更新失敗 検索エンジンの呼出しでエラーを返却	サーバ運営者に対応を依頼

7.4.3. レスポンスデータの形式

表 7-10 DB 画像更新のレスポンスデータ

カラム名	説明	備考
-	-	-

7.5. DB 画像削除

指定された画像 ID のデータを画像 DB から削除します。

7.5.1. リクエストデータの形式

メソッドタイプ:DELETE

リクエスト URL: <http://example.com/vdsapi/images/{画像 ID}>

表 7-11 DB 画像削除のリクエストデータ

カラム名	説明	備考
id	削除対象の画像 ID	

7.5.2. ステータスコード

表 7-12 DB 画像削除のステータスコード

ステータスコード	説明	備考
200	削除成功	
404	削除失敗 画像 ID に対するレコードが存在しない	
500	削除失敗 サーバ内部エラー	サーバ運営者に対応を依頼

7.5.3. レスポンスデータの形式

表 7-13 DB 画像削除のレスポンスデータ

カラム名	説明	備考
-	-	-

7.6. 画像検索

部分画像検索を行います。リクエストデータ内に以下の指定があります。

- 検索するクエリ画像はバイナリデータ
- 検索対象の画像 ID のリスト

FPGA 検索が指定時間内に完了しない場合はタイムアウト（FPGA 無応答）とします。

タイムアウト発生により各検索要求リクエストは以下の挙動となります。

表 7-14 画像検索のリクエストデータ

検索要求リクエスト	説明	備考
タイムアウト発生時の検索要求リクエスト	各検索要求リクエストの結果格納ディレクトリにエラー記録ファイルを出力する。	画像検索結果取得時にエラー（500）となる。
検索要求受付済み、かつ、検索未実行の検索要求リクエスト		
タイムアウト発生以降のアクセスによる検索要求リクエスト	検索受付不可としてエラー（503）を返す。	

Tomcat 起動時点からの画像検索の回数をカウントし、画像検索完了時点で、検索できる残回数を計算する。検索できる残回数 = $([vm.max_map_count] - [基本使用メモリマップ数^{*1}]) / [検索ごとの使用メモリマップ数^{*2}]$

検索できる残回数が"検索残回数閾値(search.map.threshold, 5.1.1 節参照)"を下回った場合は、ログに警告メッセージ(7.1.2 節参照)を出力します。¹

※ 1 : Servlet プロパティ search.map.base

※ 2 : Servlet プロパティ search.map.use

7.6.1. リクエストデータの形式

メソッドタイプ:POST

リクエスト URL: `http://example.com/vdsapi/search/ query=《クエリ画像ファイル》`

`ids=[DB 画像 ID1] [ids=[DB 画像 ID2]]...`

表 7-15 画像検索のリクエストデータ

カラム名	説明	備考
ids	検索対象画像 ID のリスト	
InputStream queryStream	クエリ画像の入カストリーム	
params	※未使用	

7.6.2. ステータスコード

表 7-16 画像検索のステータスコード

ステータスコード	説明	備考
202	検索要求登録成功	
400	検索要求登録失敗 不正なパラメータ	
500	削除失敗 サーバ内部エラー	サーバ運営者に対応を依頼
503	検索要求登録失敗 <ul style="list-style-type: none">● FPGA 無効● キューフル● タイムアウト発生による検索受付不可	

7.6.3. レスポンスデータの形式

表 7-17 画像検索のリクエストデータ

カラム名	説明	備考
id	画像検索リクエスト ID	

¹ FPGA ボードのドライバ(bw2tk)の既知障害のため、検索を 1 回行くと pmap を 2 つ確保したままになる。pmap 値が vm.max_map_count を超えるとメモリ確保できなくなり、その結果検索できなくなる。これを回避するため、このログを検出した際には手動または外部ツールを用いて Tomcat の再起動が必要となる。

7.7. 画像検索結果一覧取得

これまで画像検索依頼を行った画像検索リクエスト ID のリストを通知します。

7.7.1. リクエストデータの形式

メソッドタイプ:GET

リクエスト URL: <http://example.com/vdsapi/results/>

表 7-18 画像検索結果一覧取得のリクエストデータ

カラム名	説明	備考
-	-	

7.7.2. ステータスコード

表 7-19 画像検索結果一覧取得のステータスコード

ステータスコード	説明	備考
200	取得成功	
404	取得失敗 検索結果が存在しない	
500	取得失敗 サーバ内部エラー	サーバ運営者に対応を依頼

7.7.3. レスポンスデータの形式

表 7-20 DB 画像検索結果一覧取得のレスポンスデータ

カラム名	説明	備考
searchlist	リクエスト ID の一覧	

7.8. 画像検索結果取得

画像検索結果を通知します。

7.8.1. リクエストデータの形式

メソッドタイプ:GET

リクエスト URL: <http://example.com/vdsapi/results/{リクエスト ID}>

表 7-21 画像検索結果取得のリクエストデータ

カラム名	説明	備考
id	リクエスト ID	

7.8.2. ステータスコード

表 7-22 画像検索結果取得のステータスコード

ステータスコード	説明	備考
200	結果取得成功 画像検索処理のステータス（検索中または完了）を通知する 完了ステータスの場合は、検索結果を通知する-	
404	結果取得失敗 画像検索結果が存在しない	
500	結果取得失敗 <ul style="list-style-type: none">● タイムアウトエラー● サーバ内部エラー	サーバ運営者に対応を依頼

7.8.3. レスポンスデータの形式

表 7-23 DB 画像検索結果一覧取得のレスポンスデータ (Result)

カラム名	説明	備考
id	リクエスト ID	
accepted_at	受付日時	
finished_at	終了日時	
query_filename	クエリファイル名	
state	ステータス文字列。 検索中 or 完了	
progress	進捗率(100 以下の数値)	
message	progress + % 完了しました。	
images	画像検索結果でマッチした画像 ID のリスト	

表 7-24 List<ImageBean>の一要素のデータ形式

カラム名	説明	備考
id	画像 ID	
similarity	類似度 (0.0 から 1.0 の範囲の小数)	
origin	クエリ画像の類似箇所	
found	検索結果画像の類似箇所	

表 7-25 RectBean のデータ形式

カラム名	説明	備考
left	矩形の左上の x 座標	
top	矩形の左上の y 座標	
width	矩形の横方向 (x 軸方向) の割合 (%)	
height	矩形の縦方向 (y 軸方向) の割合 (%)	

検索結果は、servlet プロパティファイルの result.dir.path キー (5.1.1 節参照) で指定するディレクトリパス配下に、リクエスト ID をディレクトリ名として記録されます。検索結果は、result.dir.path キーで設定しているファイル領域を消費しますので、定期的に画像検索結果削除 (6.9 節参照) により画像検索結果の削除を行ってください。

7.9. 画像検索結果削除

画像検索リクエスト ID で指定された画像検索結果を削除します。

7.9.1. リクエストデータの形式

メソッドタイプ:DELETE

リクエスト URL: <http://example.com/vdsapi/results/{リクエスト ID}>

表 7-26 画像検索結果削除のリクエストデータ

カラム名	説明	備考
id	画像検索リクエスト ID	

7.9.2. ステータスコード

表 7-27 画像検索結果削除のステータスコード

ステータスコード	説明	備考
200	削除成功	
404	削除失敗 画像検索リクエスト ID に対する検索結果が存在しない	
500	削除失敗 サーバ内部エラー	サーバ運営者に対応を依頼

7.9.3. レスポンスデータの形式

表 7-28 画像検索結果削除のレスポンスデータ

カラム名	説明	備考
-	-	

8. リソース

8.1. エラーメッセージ

検索エンジンが標準出力に出力するメッセージは、Tomcat のログ管理 Catalina で収集しログファイルに保存します。

Servlet が出力するメッセージは、vdsapi_app.log で管理しログファイルに保存します。

8.1.1. 凡例

エラーメッセージの出力形式を以下に示します。

表 8-1 エラーメッセージの書式

書式				
<div style="border: 1px dashed black; padding: 10px; margin: 10px 0;"><table border="1" style="border-collapse: collapse;"><tr><td style="padding: 2px 10px;">X</td><td style="padding: 2px 10px;">0000</td><td style="padding: 2px 10px;">,</td><td style="padding: 2px 10px;">メッセージ</td></tr></table></div>	X	0000	,	メッセージ
X	0000	,	メッセージ	
X:メッセージレベル				
N: 通知レベル				
W: 警告レベル				
E: エラーレベル				
0000 : メッセージ番号				
左端の 4 桁目でモジュールを識別する				
残り 3 桁の数字は、モジュール内で一意なメッセージ番号				
, : カンマ文字(固定文字):				
メッセージ : メッセージ本文				

8.1.2. エラーメッセージ一覧

表 8-2 エラーメッセージ一覧

エラーID	メッセージ
N0001	Your image was registered successfully.
N0002	Your image was updated successfully.
N0003	Your image was deleted successfully.
E0001	An internal server error occurred.
E0002	An illegal request was sent.
E0003	Your image ID is duplicate.
E0004	Search request frequency exceeded the limitation.
E0005	The resource is not found.
E0006	The method is not allowed.
E0007	The media type is not accepted.
E0008	FPGA search is disabled.
E0009	You must provide ids by argument.
E0010	You must provide query by argument.
E0011	You must provide id by argument.
E0012	You must provide file by argument.
E0013	Your image type is not accepted.
E0014	Your image is empty.
E0015	No effective image IDs are specified.
E0016	Your image is too large. (file size)
E0017	Your image is too large. (pixel width/height)
E0018	Your image ID is too long.
E0019	Your image ID contains not allowed characters.
E0020	Registered images exceeded the limitation.
E0021	An error occurred in the search engine.
E0022	The search request can't be accepted because an error occurred in the search engine.
W0001	Warning.Number of remaining searches is【検索残回数】. There will be insufficient memory mapping area.
N1001	N1001, Start MatchingEngineManager::init.
N1002	N1002, ***** Param: 【キー名】,【キー値】
E1003	E1003, Error memory allocation in MatchingEngineManager.init.
W1004	W1004, Return code of finalFPGA() is not normal value.
N1005	N1005, End MatchingEngineManager::init.

エラーID	メッセージ
N1006	N1006, MatchingEngineManager::initFPGA.
N1008	N1008, Start MatchingEngineManager::calcSimilarFpga.
N1009	N1009, org image: w=【横幅】, h=【高さ】
N1010	N1010, new image: w=【横幅】, h=【高さ】
N1011	N1011, End MatchingEngineManager::calcSimilarFpga.
N1012	N1012, MatchingEngineManager::progressCallback2
N1013	N1013, done: 【処理済数】, sum: 【総数】
N1014	N1014, MatchingEngineManager::resultCallback2
N1015	N1015, Call information at error detection for calcSimilarFpga.
N1015	N1015, ***** 【付加情報】
E1018	E1018, Error not found messageID: 【メッセージ ID】
E1019	E1019, Error that generates Cmatching instance.
E1020	E1020, Can't load image. : 【画像ファイル名】
E1021	E1021, Error loading image. : 【画像ファイル名】, 【理由】
E1022	E1022, Error memory allocation in MatchingEngineManager.calcSimilarFpga.
E1023	E1023, Error that convert query image data. 【画像ファイル名】
E1024	E1024, Error image largest : 【画像ファイル名】, width:【横幅】, height:【高さ】
E1025	E1025, Memory allocation error in MatchingEngineManager::getBrief.
E1026	E1026, Error write progress file. : 【画像検索プログレスファイル名】
E1027	E1027, Error write result file. : 画像検索実行結果ファイル名】
E1028	E1028, Can't open inifile. :【ファイル名】
E1029	E1029, Error key value. 【キー名=【キー値】. :【理由】
W2001	W2001, Can't create more than 1 CMatching instance.
W2002	W2002, This instance of CMatching is none.
E2003	E2003, Matching pairs are nothing.
E2004	E2004, Memory allocation error in 【処理名】.
E2005	E2005, Memory allocation error for 【インスタンス名】 in 【処理名】.
W2006	W2006, resultCallback function is none.
W2007	W2007, Result for PIRFPGA_GetNextResultPointer is nothing.
W2008	W2008, Found error data from parseResult.
E2009	E2009, Error not found messageID: 【メッセージ ID】
N2010	N2010, CMatching::initFPGA()
N2011	N2011, CMatching::setTargetFeaturesFPGA()
N2012	N2012, CMatching::startFPGA
N2013	N2013, CMatching::reqFeatureCallbackFPGA
N2014	N2014, i: 【DB 画像のインデックス】, 【横幅】, 【高さ】, 【特徴点の個数】
N2015	N2015, CMatching::resultCallbackFPGA

エラーID	メッセージ
N2016	N2016, nResult, 【マッチング結果の個数】
N2017	N2017, CMatchng::parseResult[cout], scaledId, modelId, peak_x, peak_y, 【scaledId】, 【modelId】,【peak_x】, 【peak_y】
N2018	N2018, CMatching::finishCallbackFPGA
N2019	N2019, isFinishedFPGA set to 【値 1:終了, 0:検索中】
E3000	E3000, fatal error (program bug) !!
E3001	E3001,【ドライバが出力するメッセージを埋め込む】
E3002	E3002, It failed in the initialization of FPGA.
E3003	E3003, Setting to the CONT register is time-out.
E3004	E3004, Making the descriptor is failed.
E3005	E3005, The malloc function is failed.
E3006	E3006, The sem_init function is failed.
E3007	E3007, The fopen function is failed.
E3008	E3008, The strtol function is failed.
E3009	E3009, The fclose function is failed.
E3010	E3010, The sem_destory function is failed.
E3011	E3011, The sem_wait function is failed.
E3012	E3012, The sem_post function is failed.
E3013	E3013, The fflush function is failed.
E3014	E3014, The write pointer is NULL.

8.2. 定量制限

下表に示す定量制限を超える場合は、エラー事象としてメッセージ通知を行います。

表 8-3 定量制限値

項目	仕様
画像データ	以下のすべてに条件に該当する <ul style="list-style-type: none"> ・ファイルサイズは、1 GB 以下 ・画像の横幅が 8192 以下でかつ、画像の高さは 8192 以下 ・画像処理・解析を行うライブラリ:OpenCV で読み込み可能な形式であること
画像 ID	以下の文字を除く文字列。全角文字を含めることも可能 <ul style="list-style-type: none"> ・半角空白 ・改行文字 ・カンマ(,) ・タブ文字など 1 バイト制御文字