

NEC

WebOTX Application Server Express V10.2
MicroService License

インストールガイド(Linux)

ごあいさつ

このたびは、WebOTX Application Server Express MicroService Licenseをお買い上げいただき、まことにありがとうございます。

本書は、お買い上げいただいたセットの内容の確認、インストール方法や利用上の注意等を中心に構成されています。本製品をお使いになる前に、必ずお読み下さい。

以下からの説明では、WebOTX Application Serverを「WebOTX AS」と省略して表現します。

WebOTX は、日本電気株式会社の登録商標です。

Microsoft、SQL Server は、米国 Microsoft Corporation の米国およびその他の国における登録商標です。

UNIX は、The Open Group の米国ならびに他の国における登録商標です。

Oracle と Java は、Oracle Corporation およびその子会社、関連会社の米国およびその他の国における登録商標です。

MySQL は、Oracle Corporation およその子会社、関連会社の米国およびその他の国における登録商標です。

Intel は、アメリカ合衆国およびまたはその他の国における Intel Corporation の商標です。

Linux は、Linus Torvalds の米国およびその他の国における登録商標もしくは商標です。

Red Hat、OpenShift は、米国およびその他の国における Red Hat, Inc.の商標または登録商標です。

DataDirect および DataDirect Connect は、Progress Software Corporation の米国およびその他の国における商標または登録商標です。

PostgreSQL は、PostgreSQL の米国およびその他の国における商標または登録商標です。

MariaDB は、MariaDB Corporation Ab 及びその子会社、関連会社の米国及びその他の国における登録商標です。

Amazon Web Services、“Powered by Amazon Web Services”ロゴ、およびかかる資料で使用されるその他の AWS 商標は、米国その他の諸国における、Amazon.com, Inc.またはその関連会社の商標です。

Kubernetes は、The Linux Foundation の米国およびその他の国における商標または登録商標です。

This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>).

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit. (<http://www.openssl.org/>).

Docker and Docker logo are trademarks or registered trademarks of Docker, Inc. in the United States and/or other countries. Docker, Inc. and other parties may also have trademark rights in other terms used herein.

その他、記載されている会社名、製品名は、各社の登録商標または商標です。

目次

1. はじめに	1
1.1. ライセンスについて.....	1
1.2. 諸元制限.....	1
2. 使用上の条件	1
2.1. ソフトウェア条件について.....	1
2.1.1. オペレーティング・システム.....	1
2.1.2. 必要なソフトウェア.....	1
2.1.3. 対応ソフトウェア.....	2
3. リソース	4
4. インストール	5
4.1. インストール前の作業.....	5
4.1.1. SELinuxの無効化.....	5
4.1.2. Dockerの環境構築.....	5
4.1.3. マイクロサービスビルドツールのインストール.....	6
4.1.4. アプリケーションの準備.....	6
4.1.5. unzipコマンドのインストール.....	6
4.2. インストール.....	7
4.3. インストール後の作業.....	9
4.3.1. データベースを使用するための準備作業 (Java).....	9
4.4. 動作確認.....	10
5. アンインストール	12
5.1. 不要なコンテナの削除.....	12
6. 注意・制限事項	13

1. はじめに

本書では、WebOTX Application Server Express で提供しているコンテナ用にカスタマイズしたWebOTX Application Server Express MicroService Licenseについて説明します。

以下からの説明では、WebOTX Application Serverを「WebOTX AS」と省略して表現します。

1.1. ライセンスについて

WebOTX Application Server Express MicroService License は動作させるDockerコンテナ数に応じてライセンスをご購入いただくことで利用できます。

WebOTX Application Server Express MicroService License は1つのDockerコンテナで動作させる場合、1ライセンス必要となります。このため、例えば5コンテナ動作させる場合は、5ライセンスの購入が必要となります。

1.2. 諸元制限

WebOTX Application Server Express MicroService Licenseには以下の諸元制限があります。

- 同時処理数
クライアントからのリクエストの同時処理数（処理スレッド数）は100本までの制限があります。この制限は、HTTPセッション数や、利用可能なクライアント数の上限ではありません。ある時点で同時にリクエスト処理を行う上限です。対象の設定値は、「アプリケーションサーバスレッドプール」のスレッドプール最大値（max-thread-pool-size）です。

2. 使用上の条件

本章では、WebOTX ASを利用するために必要な条件について説明します。

2.1. ソフトウェア条件について

本製品がサポートするオペレーティング・システム(OS)とハードウェア、および、利用するために必要な関連ソフトウェアについて説明します。

2.1.1. オペレーティング・システム

Linux

ハードウェア	オペレーティング・システム
Intel 64	Red Hat Enterprise Linux 7 Server (7.1以降) (*1,2)

(*1)SELinux設定を有効化した場合は未サポートです。無効化にしてください。

(*2)Docker 1.12.xで動作保証している、コンテナ用にカスタマイズしたWebOTX AS Expressを提供しています。

2.1.2. 必要なソフトウェア

WebOTXシステムは、実行時にJava™ Platform, Standard EditionのSDKを必要とします。サポートするSDKバージョンは次のとおりです。

- Oracle Java SE Development Kit 8 (Update 202 以降)
- Oracle Java SE Development Kit 11 (11.0.2 以降) LTS版(※1)
- OpenJDK 8 (Update 201 以降) (※2)
- OpenJDK 11 (11.0.2 以降) (※2)

(※1) Java SE Subscription(有償契約ユーザのみ取得可能)

(※2) Red Hat リリース版をサポート

【注意事項】

• WebOTX製品は、Linuxに対応したOracle社製のJava SDKをバンドルしていますが、Java SDK自体の保守は行っていませんので、ご了承ください。

• Red Hat リリースのOpenJDKは、利用バージョンのOpenJDK Development Environmentをインストールしてください。(OpenJDK 8 : java-1.8.0-openjdk-devel , OpenJDK 11 : java-11-openjdk-devel)

• Red Hat リリースのOpenJDK 11が正式対応しているRed Hat Enterprise Linux 7は、7.6以降です。

併せて、WebOTX Application Server Express MicroService Licenseはコンテナオーケストレーションツール上での動作を想定しています。評価が完了しているコンテナオーケストレーションツールは次の通りです。

- OpenShift Container Platform v 3.9
- Kubernetes v 1.9.1

本製品のインストールにはLinux版ではなく、Windows版のWebOTX Media(※)を利用します。また、開発時にWebOTX Developerの統合開発環境(WebOTX Developer (with Developer's Studio))が必要です。

(※)WebOTX Mediaは出荷時期及び対応プラットフォームにより収録製品及びバージョンが異なるため、製品Webサイト(<https://jpn.nec.com/webotx/index.html>)もしくはWebOTX Mediaのインストールガイドにて本製品およびWebOTX Developerが収録されていることを確認してください。

2.1.3. 対応ソフトウェア

データベース (Javaアプリケーション)

WebOTX ASは、JDBC 2.0からJDBC 4.1の仕様に準拠しているJDBCドライバを介して任意のDBMSへの接続をサポートするように設計されています。アプリケーションが独自の方式でデータベース・サーバに接続、またはWebOTX ASが提供するJDBCデータソースによる接続、あるいは、WebOTXのTransactionサービス機能と連携したJTAトランザクションを使用する場合には、データベース・サーバ製品にバンドルされるJDBCドライバを入手して、セットアップしなければなりません。

WebOTX ASでは以下のJDBCドライバについて動作確認を行っております。

JDBCベンダー	JDBCドライバ・タイプ	サポートするデータベース・サーバ	備考
Oracle	Type 2、4	Oracle Database 11g Release 2 (11.2.0.4)	

		Oracle Database 12c Release 1 (12.1.0.1.0)	
		Oracle Database 12c Release 1 (12.1.0.2)	
		Oracle Database 12c Release 2 (12.2.0.1.0)	
		Oracle Database 18c (18.3.0.0.0)	
		Oracle Database 19c (19.3.0.0.0)	※
Oracle UCP	Type 2, 4	Oracle Database 11g Release 1 以降、Oracle Database 19cまで	※
Microsoft	Type 4	Microsoft SQL Server 2014	
		Microsoft SQL Server 2016	
		Microsoft SQL Server 2017	
DataDirect	Type 4	「Connect for JDBC 3.3 以降」経由によるOracle接続	
PostgreSQL Development Group	Type 4	PostgreSQL 8.1 (JDBCドライバ 8.1 Build 401) ~ PostgreSQL 11.0(JDBCドライバ 42.2.5)	
Apache Derby	Type 4	Apache Derby 10.2.2 ~ 10.11.1.2	
MariaDB	Type 4	MariaDB 10.0.24(JDBCドライバ MariaDB connector/J 2.0.2) ~ MariaDB 10.3.8(JDBCドライバ MariaDB Connector/J 2.3.0)	
Amazon Aurora	Type 4	Aurora(MySQL-Compatible) 5.6.10a (JDBCドライバ mysql-connector-java-5.1.42)	

※Oracle19cは、WebOTX V10.2リリース後に追加サポートしました。

WebOTX ASは、Java EE 7 仕様互換性テスト・パッケージ(Java EE CTS)を「Connect for JDBC 3.5」と「Oracle 11g」の組み合わせでパスしています。

その他の製品についても、例えば MySQL Connector/J 5.0など、JDBC 2.0からJDBC 4.1の仕様に準拠しているJDBCドライバであれば、WebOTX ASと連携して使用することができます。ただし、十分な評価を行ってください。

3. リソース

本章では、WebOTX Application Server Express MicroService Licenseを利用して作成するコンテナイメージ及びコンテナの消費リソースについて説明します。ただし、Dockerなどの関連ソフトウェアのディスク消費量は含まれていません。

Linux (x64) コンテナイメージ作成時の消費リソース

リソース	必要条件
メモリ	最小 640 MB、推奨 1 GB 以上
ハード ディスク	1.24 GB 以上

Linux (x64) コンテナの消費リソース (コンテナ毎)

リソース	必要条件
メモリ	最小 640 MB、推奨 1 GB 以上
ハード ディスク	64MB 以上

4. インストール

本章では、WebOTX Application Server Express MicroService Licenseのインストール方法について説明します。

4.1. インストール前の作業

インストール前に行う必要のある作業について説明します。

4.1.1. SELinuxの無効化

Red Hat Enterprise Linux (RHEL)ではLinuxカーネルにセキュリティ拡張モジュールを付加したSELinuxが組み込まれています。SELinuxは既定値設定で有効になっています。WebOTX ASは、SELinuxを有効化した場合の動作をサポートしていませんので、無効化に設定する必要があります。

SELinuxの設定状態は、デスクトップ画面から「システム設定」→「セキュリティレベル」メニューを選ぶことで確認できます。あるいは、`/etc/sysconfig/selinux`ファイルを開き、「SELINUX=」フィールドの値から確認することもできます。

4.1.2. Dockerの環境構築

Dockerのコンテナ内でWebOTX ASを動作させる場合は、事前にDockerの環境構築を行う必要があります。Dockerの環境構築を行うため、ホストOSで以下の作業を実施してください。

1. ログイン名 `root` でログインします。

```
login: root
```

2. Redhat Netrowkのextraチャンネルを有効化します。次のコマンドを実行します。

```
root> subscription-manager repos --enable=rhel-7-server-extras-rpms
```

3. Dockerをインストールします。次のコマンドを実行します。

```
root> yum install docker
```

4. コンテナイメージを作成する際にRedhat Networkへのアクセスが発生するため、必要であればDockerに対してプロキシサーバの設定を行います。まず、systemdが利用するDockerの設定ファイルの配置場所を作成します。

```
root> mkdir /etc/systemd/system/docker.service.d
```

次に、環境変数HTTP_PROXYと環境変数NO_PROXYをDockerに対して設定するために、以下の内容で`/etc/systemd/system/docker.service.d/http-proxy.conf`を作成します。

```
[Service]
Environment="HTTP_PROXY=http://[プロキシサーバ]:[ポート番号]/" "NO_PROXY=[除外対
```

```
象], /var/run/docker.sock"
```

http-proxy.confの具体例を以下に示します。

```
[Service]
Environment="HTTP_PROXY=http://proxy.example.com:8080/"
"NO_PROXY=localhost,127.0.0.0/8,::1,/var/run/docker.sock"
```

また、rootユーザに対しても同様の環境変数を設定します。

5. NetworkManagerを最新バージョンにアップデートするかファイアウォールを無効化します。NetworkManagerのバージョンが古い場合、RHEL 7にインストール済みのファイアウォールが有効化されていると正常動作しません。詳細については、Red Hat社の以下の報告を参照してください。
https://bugzilla.redhat.com/show_bug.cgi?id=1098281

ファイアウォールを無効化するには、次のコマンドを実行します。

```
root> systemctl stop firewalld.service
root> systemctl disable firewalld.service
```

6. Dockerのサービスを有効化します。次のコマンドを実行します。

```
root> systemctl start docker.service
root> systemctl enable docker.service
```

4.1.3. マイクロサービスビルドツールのインストール

マイクロサービスビルドツールをWebOTX Developerインストール環境からms-build-toolディレクトリ配下を全てLinux上の任意のディレクトリへコピーしてください。

マイクロサービスビルドツールを利用するには、WebOTX Developerが必要です。WebOTX Developer V10.2をインストールした環境を用意してください。WebOTX Developerのインストール方法については、[WebOTX Developer(with Developer's Studio) > セットアップガイド]を参照してください。

4.1.4. アプリケーションの準備

あらかじめコンパイルしパッケージングしたアプリケーション(.war)を作成しておく必要があります。

ここでは、サンプル集に付属するWebアプリケーションのサンプルHelloSample.warを使用して説明します。

あらかじめサンプル集からLinux上へHelloSample.warをコピーしてください。

サンプル集については、[Application Server > Webアプリケーション > 5. サンプル集]を参照してください。

4.1.5. unzipコマンドのインストール

作成したマイクロサービスパッケージを展開するためにunzipコマンドを使用します。unzipコマンドをインストールしていない場合、事前にインストールしておく必要があります。

4.2. インストール

Dockerコンテナ上でWebOTX ASを動作させるための手順を説明します。

1. ログイン名 root でログインします。

```
login: root
```

2. 4.1.3 でコピーしたマイクロサービスビルドツールを実行しマイクロサービスパッケージを作成します。

```
root> sh <マイクロサービスビルドツール格納ディレクトリ>/bin/ms-build-tool.sh  
--app-server-name <マイクロサービスビルドツール格納ディレクトリ>  
/app-server/webotx-as-microprofile-1.2.zip --app-file <アプリケーション格納ディ  
レクトリ>/HelloSample.war
```

カレントディレクトリにマイクロサービスパッケージmy-ms-webotx-as.zipが作成されます。

3. 作成したマイクロサービスパッケージ(my-ms-webotx-as.zip)を展開します。

```
root> mv my-ms-webotx-as.zip <任意のディレクトリ>  
root> cd <my-ms-webotx-as.zipを移動した任意のディレクトリ>  
root> unzip -d my-ms-webotx-as my-ms-webotx-as.zip
```

4. JDKをコピーします。

- Oracle社製の JDK 8をWebOTX Media製品のDVD-ROM媒体からコピーする場合

マシンのDVD-ROMドライブにLinux版の「WebOTX Media (DVD) #1」を挿入してマウントします。

```
root> mount -t iso9660 /dev/cdrom /media/cdrom
```

JDKをコピーします。

```
root> cd my-ms-webotx-as  
root> cp /media/cdrom/JDK/LINUX/jdk-*.rpm .
```

- Oracle社製の JDK 11を使用する場合

ダウンロードしたrpmファイルを上記手順 3 で展開したmy-ms-webotx-asディレクトリに配置してください。

- OpenJDK 8またはOpenJDK 11を使用する場合

yumを使用してインストールするため、ここでは何も行いません。

5. lsコマンドでファイル一覧を表示し、以下の構成になっていることを確認します。

```
root> ls -l
```

```

-rw-r--r--      1 root root          2640 Feb 12 10:59 Dockerfile
drwxr-xr-x     11 root root           144 Feb 12 10:58 WebOTX
-rw-r--r--      1 root root    178418154 Feb 12 09:47 jdk-***-linux-x64.rpm (*)
drwxr-xr-x      4 root root           44 Feb 12 10:58 share.nec

```

※:Oracle社製のJDK 8またはJDK11を使用する場合

6. Dockerfileを修正します。

- Oracle社製の JDK 8またはJDK11を使用する場合

Dockerfileの修正は不要です。

- OpenJDK 8を使用する場合

Dockerfileを以下のように修正します。

1	#	
2	# Copyright (C) 2018 NEC Corporation. All rights reserved.	
3	# Use is subject to license terms.	
4	#	
5		
6	FROM registry.access.redhat.com/rhel	
7		
8	MAINTAINER NEC Corporation	
9		
10	# install basic packages	
11	RUN yum -y install findutils	
12	RUN yum -y install procps-ng	
13	RUN yum -y install grep	
14		
15	# install JDK	
16	#COPY jdk-*.rpm /tmp/	} 行頭に # を追加しコメントアウトします。
17	#RUN rpm -ihv /tmp/jdk-*.rpm	
18	#RUN rm /tmp/jdk-*.rpm	
19		
20	RUN yum -y install java-1.8.0-openjdk-devel	} 追加します。
21	RUN mkdir -p /usr/java	
22	RUN ln -s /usr/lib/jvm/java-1.8.0 /usr/java/default	
23		
24	# install WebOTX AS Express	
25	RUN mkdir /etc/WebOTX102	

- OpenJDK 11を使用する場合

Dockerfileを以下のように修正します。

```

1 #
2 # Copyright (C) 2018 NEC Corporation. All rights reserved.
3 # Use is subject to license terms.
4 #
5
6 FROM registry.access.redhat.com/rhel
7
8 MAINTAINER NEC Corporation
9
10 # install basic packages
11 RUN yum -y install findutils
12 RUN yum -y install procps-ng
13 RUN yum -y install grep
14
15 # install JDK
16 #COPY jdk-*.rpm /tmp/
17 #RUN rpm -ihv /tmp/jdk-*.rpm
18 #RUN rm /tmp/jdk-*.rpm
19
20 RUN yum -y install java-11-openjdk-devel
21 RUN mkdir -p /usr/java
22 RUN ln -s /usr/lib/jvm/java-11 /usr/java/default
23
24 # install WebOTX AS Express
25 RUN mkdir /etc/WebOTX102

```

行頭に # を追加しコメントアウトします。

追加します。

7. Dockerのコンテナイメージを作成します。以下のコマンドを実行します。

```
root> docker build -t webotx-micro .
```

8. Dockerのコンテナイメージが作成されていることを確認します。以下のコマンドを実行します。

```

root> docker images
REPOSITORY                                TAG      IMAGE ID      CREATED
webotx-micro                              latest   *****     ** **** ago
registry.access.redhat.com/rhel          latest   *****     ** **** ago

```

(*) *** 部分の表記は環境によって異なります。

REPOSITORYの列がwebotx-microで、TAGの列がlatestであるイメージがWebOTX ASのコンテナイメージです。

これでインストール作業は完了です。
この作業が完了したら「4.3 インストール後の作業」に進んでください。

4.3. インストール後の作業

4.3.1. データベースを使用するための準備作業 (Java)

Javaアプリケーションでデータベースを使用する場合には、各データベースで次の準備作業を行ってください。詳細については、各データベースのリファレンスマニュアルでご確認ください。

Oracle での作業

- トランザクションのリカバリを行うためには、`DBA_PENDING_TRANSACTIONS` ビューの `SELECT` 権限が必要です。JDBC リソースを登録する際に、`SELECT` 権限を持つユーザを設定してください。JDBC リソースの登録を省略する場合には、トランザクション実行時に使用するJDBC データソースの定義で 指定した全ユーザに対して、`SELECT` 権限を付与してください。
- Oracle Call Interface (OCI)のJDBCドライバを使用する場合は、Oracleデータベースの環境変数を設定する必要があります。詳細は、WebOTXオンラインマニュアルの「注意制限事項 > 機能ごとの注意制限事項 > JDBCデータソース」を参考してください。
(※) JTAやTransactionサービスによるトランザクション制御を行わない場合も必要な作業です。

JDBC データソースの設定で、データベースクラスタの使用有無[`useDatabaseCluster`]に `true` を設定した場合、または、次のバージョン以降のOracleデータベースを使用する場合、ユーザアカウントに `sys.dbms_system` パッケージへの `EXECUTE`権限を付与してください。

Oracle Database 11g Release 2 (11.2.0.4)

Microsoft SQL Server での作業

- SQL Serverを使用するためには、`SqlJDBCXAUser` ロールの権限が必要です。トランザクション実行時に使用するJDBC データソースの定義で指定した全ユーザに対して、`SqlJDBCXAUser` ロールを付与してください。
- 未完了のトランザクションが存在する状態でMicrosoft SQL Server を再起動すると、Transaction サービスから データベースへの接続ができず、未完了トランザクションのリカバリを行うことができません。あらかじめ、Transaction サービスから接続するデータベースと、アプリケーションから接続するデータベースを 分けるようにしてください。例えば、Transaction サービスでリカバリを行う際に使用するデータベースを `master` とし、アプリケーションが使用するデータベースを `pubs` としてください。
- 各JDBC ドライバの分散トランザクション制御用のプログラムをインストールしてください。SQL Server JDBC Driver 3.0/ SQL Server JDBC Driver 4.0 は、SQL Server 2014 に接続することができます。SQL Server JDBC Driver 4.2 では、SQL Server 2016/ SQL Server 2017に接続することができます。

4.4. 動作確認

WebOTXの動作確認は次のとおりの手順で行ないます。

1. ホストOSにログイン名 `root` でログインします。

login: root

2. コンテナを起動します。また、コンテナ起動時にホスト8080番ポートへの通信をコンテナ80番ポートへ転送します。以下のコマンドを実行します。

```
root> docker run -it -d -p 8080:80 webotx-micro
```

Caution

「8080番ポート」はホスト上で使用していないポート番号を指定してください。
ポート番号が競合していないことを確認するには、以下のコマンドで確認できます。

```
root> ss -ant | grep 8080
```

3. 以下のコマンドを実行して、コンテナが起動していることを確認します。

```
root> docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED
537d3c548628	webotx-micro	"/opt/WebOTX/run.sh"	2minutes ago

4. HelloSampleを利用して、コンテナで動作しているWebOTX ASへ接続確認します。Webブラウザを起動し、次のURLを入力してください。

```
http://[ホストOSのIPアドレス]:8080/HelloSample/hello.html
```

以上が確認できれば、正しくインストールされています。

5. アンインストール

DockerのコンテナにインストールしたWebOTX ASについては、アンインストールはサポートしません。不要になったコンテナは、以下の手順でホストOS上から削除してください。

5.1. 不要なコンテナの削除

DockerのコンテナにインストールしたWebOTX ASについては、アンインストールはサポートしません。不要になったコンテナは、以下の手順でホストOS上から削除してください。

1. ホストOSにログイン名 `root` でログインします。

```
login: root
```

2. 不要になったコンテナが起動している場合は、ホストOS上で以下のコマンドを実行して停止してください。

```
root> docker kill [コンテナID]
```

3. 不要になったコンテナは、ホストOS上で以下のコマンドを実行して削除してください。

```
root> docker rm [コンテナID]
```

4. また、不要になったコンテナイメージは、ホストOS上で以下のコマンドを実行して削除してください。

```
root> docker rmi [イメージID]
```

6. 注意・制限事項

注意・制限事項については、オンラインマニュアルを参照して下さい。