



# Payara Server 5 管理 チート・シート

The Payara® Platform - Production-Ready,  
Cloud Native and Aggressively Compatible.

## Contents

Payara Server 5 管理 チート・シート (Japanese)	1
ドメインの起動・停止	1
スタンドアロン・インスタンスの起動・停止	1
デプロイメント・グループの起動・停止・管理	2
アプリケーションのデプロイ・アンデプロイ	4
JDBC 接続プールの作成と削除	6
モニターの有効化・無効化	7
サーバーのモニタリング	9
ログの構成と表示	11
セキュア管理を用いた DAS への接続	12

# Payara Server 5 管理 チート・シート (Japanese)

よく使われる管理操作の逆引きリファレンスです。すべてのコマンドとオプションの詳細については [Payara Server Documentation](#) を参照してください。

## ドメインの起動・停止

ユースケース	操作
ドメインの開始	<code>asadmin start-domain [domain-name] *<sup>1</sup></code>
ドメインの停止	<code>asadmin stop-domain [domain-name] *<sup>1</sup></code> 管理コンソール : Common Tasks (共通タスク) > server (Admin Server) (サーバー(管理サーバー)) > Stop (停止)
ドメインの再起動	<code>asadmin restart-domain [domain-name] *<sup>1</sup></code> 管理コンソール : Common Tasks (共通タスク) > server (Admin Server) (サーバー(管理サーバー)) > Restart (再起動)
ドメインの状態の表示	<code>asadmin list-domains</code>

\*1: *domain-name* 省略時は **domain1**

## スタンドアロン・インスタンスの起動・停止

スタンドアロン・インスタンスは以下の 2 種類:

- “ローカル” インスタンスは DAS と同じマシン上のインスタンス (基本的に ‘CONFIG’ ノードとして構成される)
- “リモート” インスタンスは DAS と異なるマシン上のインスタンスで、‘SSH’ ノードまたは ‘DCOM’ ノードとして構成される

ユースケース	操作
ローカル・インスタンスの起動	<code>asadmin start-local-instance [--sync={normal full}] instance-name *<sup>1</sup> *<sup>2</sup></code>
ローカル・インスタンスの停止	<code>asadmin stop-local-instance instance-name *<sup>1</sup></code>

ユースケース	操作
ローカル・インスタンスの再起動	<code>asadmin restart-local-instance instance-name</code> *1
インスタンスの起動 (ローカルまたはリモート)	<code>asadmin start-instance [--sync={normal full}] instance-name</code> *2 管理コンソール : Common Tasks (共通タスク) > Instances (スタンドアロン・インスタンス)
インスタンスの停止 (ローカルまたはリモート)	<code>asadmin stop-instance instance-name</code> 管理コンソール : Common Tasks (共通タスク) > Instances (スタンドアロン・インスタンス)
インスタンスの再起動 (ローカルまたはリモート)	<code>asadmin restart-instance instance-name</code> 管理コンソール : Common Tasks (共通タスク) > Instances (スタンドアロン・インスタンス)
インスタンスの状態の表示	<code>asadmin list-instances</code> 管理コンソール : Common Tasks (共通タスク) > Instances (スタンドアロン・インスタンス)

\*1: ドメインが動作していなくても実行可能

\*2: インスタンスが DAS と同期しなくなったときには --sync オプションの明示が必要 (例: `asadmin start-instance --sync=full instance1`)

## デプロイメント・グループの起動・停止・管理

ユースケース	操作
デプロイメント・グループの作成	<code>asadmin create-deployment-group deployment-group-name</code> 管理コンソール : Common Tasks (共通タスク) > Deployment Groups (デプロイメント・グループ)
デプロイメント・グループの削除	<code>asadmin delete-deployment-group deployment-group-name</code> 管理コンソール : Common Tasks (共通タスク) > Deployment Groups (デプロイメント・グループ)

ユースケース	操作
デプロイメント・グループの一覧	<pre>asadmin list-deployment-groups</pre> 管理コンソール : Common Tasks (共通タスク) > Deployment Groups (デプロイメント・グループ)
インスタンスをデプロイメント・グループへ追加	<pre>asadmin add-instance-to-deployment-group --instance instance-name --deploymentgroup deployment-group-name</pre> 管理コンソール : Common Tasks (共通タスク) > Deployment Groups (デプロイメント・グループ)
デプロイメント・グループからインスタンスを削除	<pre>asadmin remove-instance-from-deployment-group --instance instance-name --deploymentgroup deployment-group-name</pre> 管理コンソール : Common Tasks (共通タスク) > Deployment Groups (デプロイメント・グループ)
デプロイメント・グループの起動	<pre>asadmin start-deployment-group deployment-group-name</pre> 管理コンソール : Common Tasks (共通タスク) > Deployment Groups (デプロイメント・グループ)
デプロイメント・グループの再起動	<pre>asadmin restart-deployment-group deployment-group-name</pre> 管理コンソール : Common Tasks (共通タスク) > Deployment Groups (デプロイメント・グループ)
デプロイメント・グループの停止	<pre>asadmin stop-deployment-group deployment-group-name</pre> 管理コンソール : Common Tasks (共通タスク) > Deployment Groups (デプロイメント・グループ)

## アプリケーションのデプロイ・アンデプロイ

ユースケース	操作
アプリケーションのデプロイ (DAS)	<pre>asadmin deploy [--contextroot context-root] [--name=component-name] file-archive directory *1</pre> <p>管理コンソール : Common Tasks (共通タスク) &gt; Applications (アプリケーション)</p>
デプロイメント・グループへのアプリケーションのデプロイ	<pre>asadmin deploy [--contextroot context-root] [--name=component-name] --enabled=true -- target=deployment-group-name file-archive</pre> <p>管理コンソール : Common Tasks (共通タスク) &gt; Applications (アプリケーション)</p>
インスタンスへのアプリケーションのデプロイ	<pre>asadmin deploy [--contextroot context-root] [--name=component-name] --enabled=true -- target=instance-name file-archive</pre> <p>管理コンソール : Common Tasks (共通タスク) &gt; Applications (アプリケーション)</p>
インスタンスへのアプリケーションのデプロイ (既に他のインスタンスにデプロイされている場合)	<pre>asadmin create-application-ref --target= instance-name component-name</pre> <p>管理コンソール : Common Tasks (共通タスク) &gt; Applications (アプリケーション)</p>
複数のインスタンスに対するアプリケーションのデプロイ	<pre>asadmin deploy [--contextroot context-root] --name=component-name --enabled=true --target= instance-name-1 file-archive</pre> <pre>asadmin create-application-ref --target= instance-name-2 component-name *2</pre> <p>管理コンソール : Common Tasks (共通タスク) &gt; Applications (アプリケーション)</p>
アプリケーションのアンデプロイ (DAS)	<pre>asadmin undeploy [--target=server] component-name</pre> <p>管理コンソール : Common Tasks (共通タスク) &gt; Applications (アプリケーション)</p>

ユースケース	操作
デプロイメント・グループからのアプリケーションのアンデプロイ	<pre>asadmin undeploy --target=deployment-group-name component-name</pre> 管理コンソール : Common Tasks (共通タスク) > Applications (アプリケーション)
インスタンスからのアプリケーションのアンデプロイ (すべてのインスタンスから完全削除)	<pre>asadmin undeploy --target=instance-name component-name</pre> 管理コンソール : Common Tasks (共通タスク) > Applications (アプリケーション)
インスタンスからのアプリケーションのアンデプロイ (他のインスタンスではデプロイされたまま)	<pre>asadmin delete-application-ref --target=instance-name component-name</pre> 管理コンソール : Common Tasks (共通タスク) > Applications (アプリケーション)
複数のインスタンスに対するアプリケーションのアンデプロイ	<pre>asadmin delete-application-ref --target=instance-name2 component-name *2</pre> <pre>asadmin undeploy --target=instance-name1 component-name</pre> 管理コンソール : Common Tasks (共通タスク) > Applications (アプリケーション)
デプロイ済みアプリケーションの表示	<pre>asadmin list-applications [target] *3</pre> 管理コンソール : Common Tasks (共通タスク) > Applications (アプリケーション)
デプロイ済みアプリケーションの表示 (インスタンス上にあるアプリケーションのみ)	<pre>asadmin list-application-refs [target] *3</pre> 管理コンソール : Common Tasks (共通タスク) > Applications (アプリケーション)

\*1: 複数インスタンスの環境では、すべてのインスタンスが *directory* にアクセス可能でなければならぬため、*directory* の指定は推奨されない

\*2: インスタンスの数だけ繰り返す; 例: 3 インスタンスの場合は 2 回繰り返す

\*3: *target* が省略された場合はデフォルト値 **server** となる

## JDBC 接続プールの作成と削除

ユースケース	操作
JDBC ドライバのインストール	<pre>asadmin add-library [--type=app] jar-file</pre>
JDBC 接続プールの作成 (非 XA データソース)	<pre>asadmin create-jdbc-connection-pool [--data-sourceclassname=class-name] --restype=javax.sql.DataSource [-property name=value][:name=value]*] pool-name *<sup>1</sup></pre> <p>管理コンソール : Common Tasks (共通タスク) &gt; JDBC &gt; JDBC Connection Pools (JDBC接続プール) (※推奨)</p>
JDBC 接続プールの作成 (XA データソース) 例: EJB Timer 用	<pre>asadmin create-jdbc-connection-pool [--datasourceclassname=class-name] --restype=javax.sql.XADataSource [-property name=value][:name=value]*] pool-name *<sup>1</sup></pre> <p>管理コンソール : Common Tasks (共通タスク) &gt; JDBC &gt; JDBC Connection Pools (JDBC接続プール) (※推奨)</p>
JDBC 接続プールの削除	<pre>asadmin delete-jdbc-connection-pool [--cascade={false true}] pool-name</pre> <p>管理コンソール : Common Tasks (共通タスク) &gt; JDBC &gt; JDBC Connection Pools (JDBC接続プール)</p>
JDBC 接続プールへの Ping 送信	<pre>asadmin ping-connection-pool pool-name</pre> <p>管理コンソール : Common Tasks (共通タスク) &gt; JDBC &gt; JDBC Connection Pools (JDBC接続プール)</p>
JDBC 接続プールの一覧	<pre>asadmin list-jdbc-connection-pools</pre> <p>管理コンソール : Common Tasks (共通タスク) &gt; JDBC &gt; JDBC Connection Pools (JDBC接続プール)</p>
JDBC リソースの作成	<pre>asadmin create-jdbc-resources --connectionpoolid pool-name [--enabled={true false}] [--target={server domain deployment-group-name instance-name}] jndi-name</pre> <p>管理コンソール : Common Tasks (共通タスク) &gt; JDBC &gt; JDBC Resources (JDBCリソース)</p>



ユースケース	操作
JDBC リソースの削除	<pre>asadmin delete-jdbc-resources [--enabled={true false}] [--target={server domain instance-name}] jndi-name</pre> <p>管理コンソール : Common Tasks (共通タスク) &gt; JDBC &gt; JDBC Resources (JDBCリソース)</p>
JDBC リソースの一覧	<pre>asadmin list-jdbc-resources [--target={server domain instance-name}]</pre> <p>管理コンソール : Common Tasks (共通タスク) &gt; JDBC &gt; JDBC Resources (JDBCリソース)</p>

\*1:他にも多数のオプションがあり、多くがパフォーマンスに影響する

## モニターの有効化・無効化

ユースケース	操作
モニター・サービスの有効化 <sup>*1</sup>	<pre>asadmin set configs.config.\${config-name}.monitoring-service.monitoring-enabled=true</pre> <pre>asadmin set configs.config.\${config-name}.monitoring-service.mbean-enabled=true</pre> <p>管理コンソール : Configuration (構成) &gt; \${config-name} &gt; Monitoring (モニター) &gt; General (タブ) &gt; Monitoring Service (モニター・サービス) → Enabled (有効) にチェックを入れる</p> <p>管理コンソール : Configuration (構成) &gt; \${config-name} &gt; Monitoring (モニター) &gt; General (タブ) &gt; Monitoring MBeans (モニターMBean) → Enabled (有効) にチェックを入れる</p>

ユースケース	操作
モニター・サービスの無効化 <sup>*1</sup>	<pre>asadmin set configs.config.\${config-name}. monitoring-service.mbean-enabled=false</pre> <pre>asadmin set configs.config.\${config-name}. monitoring-service.monitoring-enabled=false</pre> <p>管理コンソール : Configuration (構成) &gt; \${config-name} &gt; Monitoring (モニター) &gt; General (タブ) &gt; Monitoring Service (モニター・サービス) → Enabled (有効) のチェックを外す</p> <p>管理コンソール : Configuration (構成) &gt; \${config-name} &gt; Monitoring (モニター) &gt; General (タブ) &gt; Monitoring MBeans (モニターMBean) → Enabled (有効) のチェックを外す</p>
モニター・レベルの変更 (一括) <sup>*1*2</sup>	<p>管理コンソール : Configuration (構成) &gt; \${config-name} &gt; Monitoring (モニター) &gt; General (タブ) &gt; Component Level Settings (コンポーネント・レベルの設定);</p> <ol style="list-style-type: none"> <li>1. すべてのモジュールを選択する</li> <li>2. Level (レベル) → "HIGH" ("高") (有効) または "OFF" ("オフ") (無効) を選択後 "Change Level" ("レベルの変更") をクリックする</li> <li>3. "Save" (保存) をクリックする</li> </ol>
モニター・レベルの変更 (コンポーネント単位) <sup>*1*2</sup>	<pre>asadmin set configs.config.\${config-name}. monitoring-service.module-monitoring- levels.\${module-name}=HIGH (有効)</pre> <p>または</p> <pre>asadmin set configs.config.\${config-name}. monitoring-service.module-monitoring- levels.\${module-name}=OFF (無効)</pre> <p>管理コンソール : Configuration (構成) &gt; \${config-name} &gt; Monitoring (モニター) &gt; General (タブ) &gt; Component Level Settings (コンポーネント・レベルの設定);</p> <ol style="list-style-type: none"> <li>1. \${module-name} を選択する</li> <li>2. \${module-name} &gt; Monitoring Level (モニター・レベル) → "HIGH" ("高") (有効) または "OFF" ("オフ") (無効)</li> <li>3. "Save" をクリックする</li> </ol>

ユースケース	操作
モニター設定の一覧 *1	<pre>asadmin get configs.config.\${config-name}. monitoring-service.*</pre> <p>管理コンソール : Configuration (構成) &gt; \${config-name} &gt; Monitoring (モニター) &gt; General (タブ) &gt; Component Level Settings (コンポーネント・レベルの設定)</p>

\*1: \${config-name} の例: "server-config".

\*2: \${module-name} については "表 1. コンポーネントとサービス" を参照のこと

## サーバーのモニタリング

ユースケース	操作
サーバーのモニタリング *1	<pre>asadmin get server.\${module-name}.* *2</pre> <p>管理コンソール : Common Tasks (共通タスク) &gt; server (Admin Server) (サーバー (管理サーバー)) &gt; Monitor (モニター) (タブ) &gt; Server (サーバー) (タブ)</p> <p>JMX または MicroProfile Metrics を使用する *3</p>

\*1: モニタリングを有効にするためには、1 つ以上のコンポーネント/サービスのモニターを有効化する必要がある

\*2: \${module-name} については "表 1. コンポーネントとサービス" を参照のこと

\*3: 追加の設定が必要 (環境により異なる) ため、詳細については Payara Server Documentation および Payara Blog を参照のこと

表 1. コンポーネントとサービス

asadmin..	管理コンソール..	asadmin..	管理コンソール..
jvm	Jvm	jpa	Java Persistence (Java永続性)
transaction-service	Transaction Service (トランザクション・サービス)	jdbc-connection-pool	Jdbc Connection Pool (JDBC接続プール)
connector-service	Connector Service (コネクタ・サービス)	thread-pool	Thread Pool (スレッド・プール)
jms-service	Jms Service (JMSサービス)	ejb-container	Ejb Container (EJB コンテナ)
security	Security Service (セキュリティ)	orb	ORB (Object Request Broker)
web-container	Web Container (Webコンテナ)	connector-connection-pool	Connector Connection Pool (コネクタ接続プール)
jersey	Jersey (Restful Web Services) (Jersey (RESTful Webサービス))	deployment	Deployment (デプロイメント)
web-services-container	Web Services Container (Webサービス・コンテナ)	http-service	Http Service (HTTPサービス)

## ログの構成と表示

ユースケース	操作
サーバー・ログ設定の表示と構成	<pre>                     \${install-dir}/glassfish/domains/\${domain-name}/                     config/logging.properties ファイルを直接編集する                     (※非推奨)                 </pre> <p>管理コンソール : Configuration (構成) &gt; \${config-name} &gt; Logger Settings (ロガーの設定) &gt; General (一般) (タブ)</p>
サーバー・ログの表示	<p>管理コンソール : Common Tasks (共通タスク) &gt; server (Admin Server) (サーバー(管理サーバー)) &gt; General (一般) (タブ) &gt; "View Log Files" ("ログ・ファイルの表示") をクリックする (Log Viewer)</p> <p>管理コンソール : Common Tasks (共通タスク) &gt; server (Admin Server) (サーバー(管理サーバー)) &gt; General (一般) (タブ) &gt; "View Raw Log" ("rawログの表示") をクリックする (Raw Log Viewer)</p> <p>"server.log" ファイルを直接開く (既定の場所は下記の通り):</p> <ul style="list-style-type: none"> <li>• DAS: <code>\${install-dir}/glassfish/domains/\${domain-name}/logs/server.log</code></li> <li>• Instance: <code>\${install-dir}/glassfish/nodes/\${node-name}/\${instance-name}/logs/server.log</code></li> </ul>
ログのローテーション (強制)	<pre>                     asadmin rotate-log                 </pre> <p>管理コンソール : Common Tasks (共通タスク) &gt; server (Admin Server) (サーバー(管理サーバー)) &gt; General (一般) (タブ) &gt; "Rotate Log" ("ログのローテーション") をクリックする</p>
アクセス・ログの有効化 * <sup>1</sup>	<pre>                     asadmin set configs.config.\${config-name}.http-                     service.access-logging-enabled=true                 </pre> <p>管理コンソール : Configuration (構成) &gt; \${config-name} &gt; HTTP Service (HTTPサービス) &gt; Access Logging (アクセス・ロギング) → "Enabled" ("有効") にチェックを入れる</p>

ユースケース	操作
アクセス・ログの無効化 *1	<pre>asadmin set configs.config.\${config-name}.http-service.access-logging-enabled=true</pre> <p>管理コンソール : Configuration (構成) &gt; \${config-name} &gt; HTTP Service (HTTPサービス) &gt; Access Logging (アクセス・ロギング) → “Enabled” (“有効”) のチェックを外す</p>
アクセス・ログの表示	<p>“access.log” ファイルを直接開く (既定の場所は下記の通り):</p> <ul style="list-style-type: none"> <li>• DAS: \${install-dir}/glassfish/domains/\${domain-name}/logs/access/server_access_log.YYYY-MM-DD.txt</li> <li>• Instance: \${install-dir}/glassfish/nodes/\${node-name}/\${instance-name}/logs/access/server_access_log.YYYY-MM-DD.txt</li> </ul>

\*1: デフォルトではアクセスログは無効化されている

## セキュア管理を用いた DAS への接続

ユースケース	マシン	操作
セキュア管理の設定	ローカル	<ol style="list-style-type: none"> <li>1. <code>asadmin start-domain [--domain-name]</code></li> <li>2. <code>asadmin change-admin-password</code></li> <li>3. <code>asadmin enable-secure-admin</code></li> <li>4. <code>asadmin restart-domain</code></li> </ol> <p>管理コンソール</p> <ol style="list-style-type: none"> <li>1. Common Tasks (共通タスク) &gt; Domain (ドメイン) &gt; Administrator Password (管理者パスワード) (タブ) → “New Password” (“新規パスワード”) を設定して “Save” (“保存”) をクリックする</li> <li>2. Common Tasks (共通タスク) &gt; server (Admin Server) (サーバー(管理サーバー)) &gt; “Secure Administration...” (“セキュリティ保護された管理...”) をクリックする &gt; “Enable Secure Admin” (“セキュリティ保護された管理の有効化”) をクリックする</li> <li>3. ドメイン再起動</li> </ol>

ユースケース	マシン	操作
管理コンソールの使用	ローカル	https://localhost:4848/ を開く *1 管理ユーザー名とパスワードを入力
	リモート	https://das-hostname:4848/ を開く *1 管理ユーザー名とパスワードを入力
asadmin コマンドの使用 (ログイン前)	ローカル	asadmin [--host localhost] [--user admin-user-name] command options *2 管理ユーザー名とパスワードを入力
	リモート	asadmin --host das-hostname [--user admin-user-name] command options *2 管理ユーザー名とパスワードを入力
ログイン	ローカル	asadmin [--host localhost] [--user admin-user-name] login *2 管理ユーザー名とパスワードを入力
	リモート	asadmin --host das-hostname [--user admin-user-name] login *2 管理ユーザー名とパスワードを入力
ログイン後の asadmin コマンドの使用	ローカル	asadmin [--host localhost] [--user admin-user-name] command options *2 管理ユーザー名とパスワードは入力不要
	リモート	asadmin --host das-hostname [--user admin-user-name] command options *2 管理ユーザー名とパスワードは入力不要

\*1: セキュア管理有効時は必ず "HTTPS" アクセスとなる

\*2: デフォルトでは、admin-user-name は "admin"



**[sales@payara.fish](mailto:sales@payara.fish)**



**+44 207 754 0481**



**[www.payara.fish](http://www.payara.fish)**

Payara Services Ltd 2019 All Rights Reserved. Registered in England and Wales; Registration Number 09998946  
Registered Office: Malvern Hills Science Park, Geraldine Road, Malvern, United Kingdom, WR14 3SZ