

JEUS ノードマネージャガイド

JEUS v8.0



Copyright © 2016 TmaxSoft Co., Ltd. All Rights Reserved.

Copyright Notice

Copyright © 2016 TmaxSoft Co., Ltd. All Rights Reserved.

45, Jeongjail-ro, Bundang-gu, Seongnam-si, Gyeonggi-do, 13613, South Korea

Restricted Rights Legend

All TmaxSoft Software (JEUS®) and documents are protected by copyright laws and international convention. TmaxSoft software and documents are made available under the terms of the TmaxSoft License Agreement and this document may only be distributed or copied in accordance with the terms of this agreement. No part of this document may be transmitted, copied, deployed, or reproduced in any form or by any means, electronic, mechanical, or optical, without the prior written consent of TmaxSoft Co., Ltd.

Nothing in this software document and agreement constitutes a transfer of intellectual property rights regardless of whether or not such rights are registered) or any rights to TmaxSoft trademarks, logos, or any other brand features. This document is for information purposes only. The company assumes no direct or indirect responsibilities for the contents of this document, and does not guarantee that the information contained in this document satisfies certain legal or commercial conditions. The information contained in this document is subject to change without prior notice due to product upgrades or updates. The company assumes no liability for any errors in this document.

このソフトウェア(JEUS®)マニュアルの内容とプログラムは、日本国の著作権法および国際条約によって保護されています。マニュアルの内容とプログラムは、TmaxSoft Co., Ltd.との使用許諾契約書の下でのみ使用することができ、マニュアルは使用許諾契約で許可されている範囲を除いては、配布または複製することができません。TmaxSoftの書面による事前の承諾を得ることなく、このマニュアルの全部または一部を電子的または機械的な方法を問わず、転送、複製、配布したり、または二次的著作物を作成する等の行為を一切禁じます。

このソフトウェアのマニュアルとプログラムの使用許諾契約は、いかなる場合においても、マニュアル及びプログラムと関連する知的財産権(登録の有無を問わず)を譲渡するものと解釈されず、TmaxSoftのブランド、ロゴ、商標等の使用権限を与えるものではありません。マニュアルは、情報を提供する目的でのみ提供しており、これに伴う契約上の直接的ないしは間接的な責任を負わず、マニュアルの内容は法律上もしくは商業的な特定の条件が満たされることを保証しません。マニュアルの内容は、製品のアップグレード及び修正により、その内容が予告なく変更されることがあり、内容上の誤りが無いことを保証しません。

Trademarks

JEUS® is registered trademark of TmaxSoft Co., Ltd.

JEUS®は、TmaxSoft Co., Ltd.の登録商標です。

Java and Solaris are registered trademarks of Oracle Corporation and its subsidiaries and affiliates.

Java、Solarisは、Oracle Corporation及びその子会社、関連会社の登録商標です。

Microsoft, Windows, and Windows NT are registered trademarks or trademarks of Microsoft Corporation.

Microsoft、Windows、Windows NTは、Microsoft Corporationの登録商標または商標です。

HP-UX is a registered trademark of Hewlett Packard Enterprise Company.

HP-UXは、Hewlett Packard Enterprise Companyの登録商標です。

AIX is a registered trademark of International Business Machines Corporation.

AIXは、International Business Machines Corporationの登録商標です。

UNIX is a registered trademark of X/Open Company, Ltd.

UNIXは、X/Open Company, Ltd.の登録商標です。

Linux is a registered trademark of Linus Torvalds.

Linuxは、Linus Torvaldsの登録商標です。

Other products and company names are trademarks or registered trademarks of their respective owners.

その他、記載されている会社名、製品名などは、各社の商号、商標または登録商標です。

The names of companies, systems, and products mentioned in this manual may not necessarily be indicated with a trademark symbol (TM, ®).

本マニュアルに記載されている会社名、システム名、製品名などには必ずしも商標表示(TM、®)を付記しておりません。

Open Source Software Notice

Some modules or files of this product are subject to the terms of the following licenses. : APACHE2.0, CDDL1.0, EDL1.0, OPEN SYMPHONY SOFTWARE1.1, TRILEAD-SSH2, Bouncy Castle, BSD, MIT, SIL OPEN FONT1.1

Detailed Information related to the license can be found in the following directory : \${INSTALL_PATH}/lib/licenses

この製品の一部ファイルまたはモジュールは、APACHE2.0、CDDL1.0、EDL1.0、OPEN SYMPHONY SOFTWARE1.1、TRILEAD-SSH2、Bouncy Castle、BSD、MIT、SIL OPEN FONT1.1のライセンスに準拠します。

文書情報

文書名: JEUS ノードマネージャガイド

発行日: 2016年10月14日

ソフトウェアバージョン: JEUS v8.0

ガイドバージョン: v2.1.1

目次

このガイドについて	ix
第1章 はじめに	1
1.1. 概要	1
1.1.1. 概念	1
1.1.2. 目的	2
1.1.3. 種類	2
1.1.4. 制限事項	3
1.2. タイプ別のノード・マネージャーの設定	4
1.3. 共通機能	6
1.4. RQSプロセス管理機能	7
第2章 Javaタイプのノード・マネージャー	9
2.1. 概要	9
2.2. 機能別の動作方式	9
2.2.1. リモート・マシンに存在するMS(ManagedServer)の起動	9
2.2.2. サーバー・モニタリング	10
2.2.3. 異常なサーバーの再起動	12
2.2.4. ローリング・パッチ	12
2.2.5. DASとの接続	13
2.3. 環境設定	14
2.3.1. 設定ファイル	14
2.3.2. 必須ファイル	16
2.4. ノードの設定および削除	17
2.4.1. Javaタイプのノード設定	18
2.4.2. Javaタイプのノード削除	20
2.5. ノード・マネージャーの開始および終了	21
2.6. Javaタイプのノード・マネージャーを使用したサーバーの制御	23
2.7. ログ・ファイル	26
第3章 SSHタイプのノード・マネージャー	27
3.1. 概要	37
3.2. 環境設定	27
3.2.1. SSHの設定	27
3.3. ノードの設定および削除	29
3.3.1. SSHタイプのノード設定	29
3.3.2. SSHタイプのノード変更	34
3.3.3. SSHタイプのノード削除	35
第4章 RQSプロセス管理	37
4.1. 概要	37
4.2. RQSの動作方式	37
4.2.1. RQSプロセスの起動と停止	37

4.2.2. RQSプロセスのモニタリング	38
4.2.3. 異常のプロセスの再起動	38
4.3. 環境設定	38
4.3.1. 設定ファイル	38
第5章 ノード・マネージャーの冗長化	41
5.1. 概要	41
5.2. 冗長化されたノード・マネージャーの動作方式	41
5.3. 冗長化されたノード・マネージャーの使用方法	42
5.4. 冗長化構成時のノード・マネージャーの終了	43
索引	45

図目次

[図 1.1]	JEUSとドメイン、ノード・マネージャーの関係	1
[図 1.2]	ノード・マネージャーの設定メニュー	4
[図 1.3]	タイプ別のノード・マネージャーの設定	5
[図 2.1]	ノード・マネージャーでリモート・マシンのサーバー起動	9
[図 2.2]	Javaタイプのノード・マネージャーでのサーバー・モニタリング	10
[図 2.3]	Javaタイプのノード・マネージャーのサーバー再起動	12
[図 3.1]	リモート・ノードへのSSH接続	28
[図 5.1]	冗長化されたノード・マネージャーの動作方式	41

このガイドについて

対象読者

本書では、JEUS[®] (以下、JEUS)ノード・マネージャーの概念と動作方式を理解するための基本的な内容を説明します。ドメインを設定、制御するには構造についての理解が必要です。本書は、JEUSノード・マネージャーを使用する開発者と管理者を対象としています。

前提知識

JEUSの基本的な使用方法と製品を理解するには、以下のガイドについてあらかじめ熟知することをお勧めします。

- JEUS 紹介ガイド
- JEUS インストール & スタートガイド

本書のすべてのサンプルと環境構成は、UNIXスタイルに準拠します。Microsoft Windows[™](以下、Windows)など他の環境で作業を行う場合は、次のような事項を考慮してください。

たとえば、Windowsプラットフォームでは、ディレクトリー区切り子をUNIXスタイルのスラッシュ(/)からWindowsスタイルのバックスラッシュ(\)に変えて使用してください。また、環境変数もWindowsスタイル(%%)に変更して使用してください。

本書で触れているJEUS_HOMEは、JEUSがインストールされているディレクトリーです。

制限事項

本書の内容は、Java標準に準拠して作成されていますが、本書で触れているJava EEやJava仕様については詳しく取り上げていません。関連内容についてはJava関連ドキュメントを参照してください。

本書の構成

本書は、計5章で構成されています。

- [「第1章 はじめに」](#)

ノード・マネージャーの概念、目的、種類などの基本的な内容および2タイプのノード・マネージャーの共通機能について説明します。

- [「第2章 Javaタイプのノード・マネージャー」](#)

Javaタイプのノード・マネージャーを使用するための設定方法および動作方式について説明します。

- [「第3章 SSHタイプのノード・マネージャー」](#)

SSHタイプのノード・マネージャーを使用するための設定方法について説明します。

- [「第4章 RQSプロセス管理」](#)

RQSプロセスの管理機能について説明します。

- [「第5章 ノード・マネージャーの冗長化」](#)

ノード・マネージャーが異常終了した際にノード・マネージャーの復旧を可能にする冗長化機能について説明します。

表記上の規則

表記	意味
<<AaBbCc123>>	プログラム・ソースコードのファイル名
<Ctrl>+C	CtrlキーとCキーを同時に押す
[Button]	GUIのボタン、メニュー名
太字	強調
「」、『』（鍵カッコ）	関連文書、あるいはガイド内の他の章および節の表示
「入力項目」	画面UI上の入力項目
ハイパーリンク	メール・アカウント、Webサイト
>	メニューの実行順
+----	下位ディレクトリー/ファイル有り
----	下位ディレクトリー/ファイル無し
参考	参照/注意事項
注	注意事項
[図 1.1]	図の名前
[表 1.1]	表の名前
AaBbCc123	Javaコード、XMLドキュメント
[<i>command argument</i>]	オプション・パラメータ
< xyz >	「<」と「>」の間の内容は実際に使用される特定の名前または値で置き換えられる
	構文の中の相互に排他的な選択項目の選択肢を示す 例) A B: AとBのいずれかを選択
...	パラメータ、値、または他の情報が繰り返される
\${ }	環境変数

システム要件

	要求事項
プラットフォーム	Solaris 9, 10, 11
	HP-UX 11.x, 11i, 11iV2
	IBM AIX 5L, 6L, AIX 7L
	MS Windows 2008, 2012, Vista, 7, 8
ハードウェア	最小2GB以上、推奨20GBのハードディスク容量
	推奨1GB以上のメモリー容量
JDK	JDK 7, JDK 8

関連文書

ガイド	説明
JEUS 紹介ガイド	JEUSサーバーについて全般的に紹介し、JEUSのアーキテクチャーを含む各構成要素について記述しています
JEUS インストール & スタートガイド	JEUSについて紹介し、JEUSのインストールおよび開始方法について記述しています
JEUS サーバガイド	JEUSシステムおよびサーバーの概要とシステムの管理方法について記述しています
JEUS リファレンスガイド	JEUSを使用するために必要な詳細設定とJEUSの使用方法について記述しています
JEUS WebAdmin ガイド	JEUSのWeb管理ツールであるWebAdminを利用したJEUSの設定および制御、モニタリング、クラスタリング、リソースの設定および管理について記述しています

お問合せ先

Korea

TmaxSoft Co., Ltd.
45, Jeongjail-ro, Bundang-gu,
Seongnam-si, Gyeonggi-do, 13613
South Korea
Tel: +82-31-8018-1000
Fax: +82-31-8018-1115
Email: info@tmax.co.kr
Web (Korean): <http://www.tmaxsoft.com>
TechNet: <http://technet.tmaxsoft.com>

USA

TmaxSoft Inc.
101 North Wacker Drive, Suite 2014,
Chicago, IL 60606
U.S.A
Tel: +1-312-525-8330
Email: info@tmaxsoft.com
Web (English): http://www.tmaxsoft.com/us_en/home

Japan

TmaxSoft Japan Co., Ltd.
5F Sanko Bldg, 3-12-16 Mita, Minato-Ku, Tokyo, 108-0073
Japan
Tel: +81-3-5765-2550
Fax: +81-3-5765-2567
Email: info@tmaxsoft.co.jp
Web (Japanese): <http://www.tmaxsoft.co.jp>

China

Beijing TmaxSoft System Software Co., Ltd.
Room103, No.2 Huizhong Building, Seven Street Shangdi,
Haidian District, Beijing, 100085
P.R.China
Tel: +86-10-6298-8827
Email: info@tmaxsoft.com.cn
Web (Chinese): http://www.tmaxsoft.com/cn_en/home_cn_en

Brazil

Tmax Brasil Sistemas e Serviços Ltda.
Av. Copacabana, 177, sala 32~35 Empresarial 18 do Fortel
Alphaville Barueri, Sao Paulo, 06472-001
Brazil
Tel: +55-11-4191-3100
Fax: +55(11) 4191-3705 (extension#112)
Email: info.bra@tmaxsoft.com
Web (Portuguese): http://www.tmaxsoft.com/br_en/home_br_en

Russia

Tmax Rus L.L.C.
Leninsky prospekt, 113/1 (Park Place Moscow),
Office 318e, Moscow, 117198
Russia
Tel: +7(495)970-01-35
Email: info.rus@tmaxsoft.com
Web (Russian): http://www.tmaxsoft.com/ru_ru/home_ru_ru

Singapore

Tmax Singapore Pte. Ltd.
430 Lorong 6, Toa Payoh #10-02,
OrangeTee Building, 319402
Singapore
Tel: +65-6259-7223
Fax: +65-6258-7112
Email: info.sg@tmaxsoft.com
Web (English): http://www.tmaxsoft.com/sg_en/home_sg_en

United Kingdom

TmaxSoft UK Ltd.
215 Knyvett House, Watermans Business Park,
The Causeway, Staines TW18 3BAB
United Kingdom
Tel: +44-1784-895005
Email: info.uk@tmaxsoft.com
Web (English): http://www.tmaxsoft.com/gb_en/home_gb_en

Canada

TmaxSoft Canada, Inc.
2425 Matheson Blvd East, 8th floor,
Unit 824 Mississauga, ON, L4W 5K4
Canada
Tel: +1-905-361-2888
Email: info.canada@tmaxsoft.com
Web (English): http://www.tmaxsoft.com/ca_en/home_ca_en

Australia

TmaxSoft Proprietary Limited
L32, 101 Miller Street, North Sydney 2060
Australia
Tel: +91-9845-330-704
Email: info.aus@tmaxsoft.com
Web (English): http://www.tmaxsoft.com/au_en/home_au_en

India

TmaxSoft Technologies Private Limited
Sobha Alexander Plaza, 3rd Floor,
16/2 Commissariat Road, Bangalore-560025
India
Tel: +91-9845-330-704
Email: info.india@tmaxsoft.com
Web (English): http://www.tmaxsoft.com/in_en/home_in_en

Turkey

TmaxSoft Co., Ltd. Turkey Liaison Office
Windowist Tower. Eski Buyukdere Cad. No:26,
Maslak 34467 Istanbul
Turkey
Tel: +90-544-553-6045
Email: cslee@tmaxsoft.com
Web (English): http://www.tmaxsoft.com/tr_en/home_tr_en

第1章 はじめに

本章では、ノード・マネージャーの基本情報と共通機能について説明します。

1.1. 概要

本節では、ノード・マネージャーの概念、目的、種類、ノード・マネージャーを構成する際の制限事項について説明します。

1.1.1. 概念

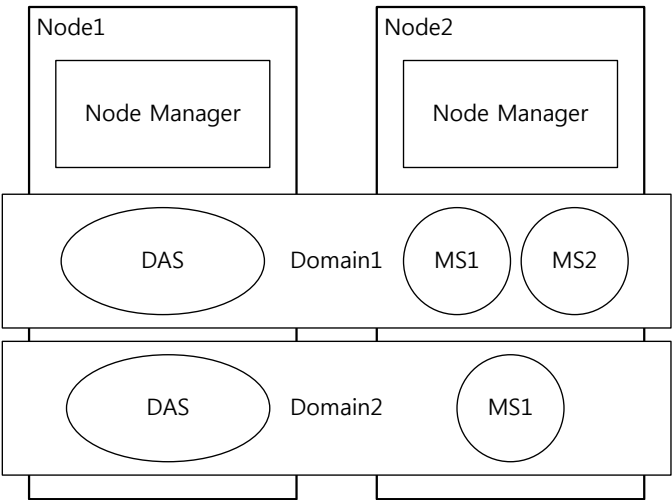
ドメイン環境では、ドメインを構成するサーバーが複数のマシンに存在することができ、1つのマシンに複数のドメインが存在することもできます。1つのマシンには複数のJEUSがインストールされ、各JEUSには1つのノード・マネージャーが存在します。

このような環境において、ドメインに限らず、1つのマシンでサーバー・プロセスを管理するのがノード・マネージャーです。すなわち、ドメイン単位ではなくマシン単位でサーバーを管理し、マシンに存在するサーバーを開始、終了、再起動して、サーバーが正常状態でサービスできるようにします。ノード・マネージャーは、各JEUSに1つずつ存在します。

サーバーが実行されるマシンにインストールされたJEUSをノードと定義します。マシンのアドレス、JEUSのインストール・ディレクトリー・パスなどの情報を利用して一意のノードが設定できます。

以下は、ノード(マシンにインストールされたJEUS)とドメイン、ノード・マネージャーの関係を表した図です。

[図 1.1] JEUSとドメイン、ノード・マネージャーの関係



1.1.2. 目的

ノード・マネージャーの主な目的は、1つのノードで動作するサーバー・プロセスを管理することです。

ノード・マネージャーは、Domain Administration Server(以下、DAS)と別のリモート・マシンで動作するサーバーを開始することができます。また、サーバーが異常終了するか、ユーザーが設定した異常状態になったときに再起動できます。サーバーが突然ダウンした場合、DASがSCF(System Clustering Framework)を利用してこれを検知し、自動的に再起動する機能を提供しています。

1ノードあたり1つのインスタンスが存在し、サーバーに適用するパッチ・ファイルを取得して適用することができます。さらに、ノード・マネージャーを利用して、DASによって管理されるサーバーをコンソール・ツール(jeusadmin)のstart-serverコマンドやJEUS WebAdmin(以下、WebAdmin)を使用して簡単に実行することができます。また、サーバー・プロセスと共に外部プロセスを管理する機能も持ちます。現在はRQSプロセスを管理する機能を提供します。RQSプロセスの起動と停止、また異常終了時に再開を行うことができます。この機能はサーバー・プロセスの管理とは独立した機能であり、設定ファイルの設定情報だけでプロセスの管理を行います。

参考

ノード・マネージャーは付加的な機能なので、使用しなくてもサーバーのサービスには影響を与えません。しかし、ドメインの安定的な運用のためにノード・マネージャーを使用することをお勧めします。

1.1.3. 種類

JEUSで提供するノード・マネージャーは以下の2種類に分けられます。

- Javaタイプのノード・マネージャー

Javaで実装されたノード・マネージャーであり、OSを問わず実行できます。

DASと別のノードに存在するManaged Server(以下、MS)を開始することができます。ノード・マネージャーはサーバーの状態と異常終了を検知し、問題が発生したサーバーの再起動動作をSSHタイプのノード・マネージャーに比べ迅速かつ簡単に行います。また、ドメインにパッチ・ファイルを適用することができます。

ただし、Javaタイプのノード・マネージャーを使用するためには、JEUSがインストールされている必要があるため、SSHタイプのノード・マネージャーとは違って別のマシンにJEUSをインストールすることができません。Javaタイプのノード・マネージャーについての詳細については、「[第2章 Javaタイプのノード・マネージャー](#)」を参照してください。

- SSHタイプのノード・マネージャー

OSから提供するSSHを使用するノード・マネージャーであり、OSがWindowsの場合には使用できません。Windowsがインストールされているマシンでは、Javaタイプのノード・マネージャーを使用してください。

SSHタイプのノード・マネージャーはJavaタイプのノード・マネージャーと同様、リモート・マシンに存在するサーバーを開始したり、異常終了を検知して再起動したりします。ただし、SSHタイプのノード・マネージャー

はサーバー・プロセスを直接モニタリングしないため、DASでSCFを利用してサーバーの異常状態を検知し、SSHタイプのノード・マネージャーを使用してリモート・サーバーを再起動します。

SSHタイプのノード・マネージャーがJavaタイプのノード・マネージャーと異なる点は、SSHタイプのノード・マネージャーを利用して別のマシンにJEUSがインストールできるということです。SSHタイプのノード・マネージャーについての詳細については、「[第3章 SSHタイプのノード・マネージャー](#)」を参照してください。

1.1.4. 制限事項

以下は、ノード・マネージャーを構成するときの制限事項です。

- ドメイン内で使用されるノード・マネージャーの名前は一意である必要があります。
- インストールした1つのJEUS内には、1つのノード・マネージャーのみ存在します。
- OSがWindowsの場合はSSHが提供されないため、Javaタイプのノード・マネージャーを使用します。
- Javaタイプのノード・マネージャーは、OSで提供するサービスに登録して使用することをお勧めします。
- Javaタイプのノード・マネージャーを使用する場合は、jeusnm.xmlファイルの「**useNodeManager**」項目をtrueに設定します。

1.2. タイプ別のノード・マネージャーの設定

JEUSでは、WebAdminとコンソール・ツールを利用してノード・マネージャーを設定できます。ノードに関連するすべての事項は、WebAdminのメイン画面の右側にある[ノード設定]メニューで編集および動作することができます。

[図 1.2] ノード・マネージャーの設定メニュー

Servers Failed 0 | Standby 0 | Running 1 | Shutdown 1 | Suspended 0 | Other 0

Server	Status	Engine
adminServer(*)	RUNNING(00:12:52)	
server1	SHUTDOWN	

DataSource Database 0 | Cluster-ds 0

Applications Application 0

Clusters Clusters 0

ドメイン基本情報

domain1

✧ ドメインセキュリティ領域設定 [設定](#)

✧ ユーザおよびグループ管理 [管理](#)

✧ オートロック環境設定 [設定](#)

オンラインヘルプ

JEUS8 WebAdminユーザマニュアルを照会します。

Webコンソールの実行

ノード設定

Resource Manager

Message Bridge
Mail Source
External Source
DataSource

Monitoring

✧ Thread	Monitoring
✧ Transaction	Monitoring
✧ MBean	Monitoring
✧ Servers	Monitoring
✧ JNDI	List

[Nodes]画面で使用するノードに応じて「Ssh」または「Java」を選択して設定します。

[図 1.3] タイプ別のノード・マネージャーの設定

Nodes

HISTORY

ノードを設定します。

ヘルプ

確認再設定

Name *	node1 一意のノード名を指定します。		
Host *	localhost	EX	192.168.1.xxx ノードのホスト名を指定します。
Mapped Servers	adminServer server1 当該ノードで管理するサーバの名前を表示します。		

Ssh

ノードのSSHタイプのノードマネージャ設定を定義します。

Install Dir		EX	/home/user/jeus ノードのJEUインストールディレクトリを指定します。
User		SSHノードの認証に使用されるユーザ名を指定します。デフォルト値はシステムユーザ名です。	
Private Key File		EX	/home/user/.ssh/id_rsa SSHノードの認証に使用される個人キーファイルを指定します。デフォルト値はシステムユーザのホームディレクトリ配下の"/.ssh/id_rsa"です。
Port		SSHノードのポート番号を指定します。デフォルト値は22番ポートです。	

Java

ノードのJavaタイプのノードマネージャ設定を定義します。

Port	7730	[デフォルト: 7730] ノードマネージャで使用するポートを指定します。デフォルト値は7730です。	
Use Ssl	<input type="checkbox"/>	[デフォルト: false] ノードマネージャでSSLを使用するか否かを設定します。	
Truststore Path		ノードマネージャでSSLを使用する場合、認証に使用されるトラストストアファイルのパスを指定します。	
Truststore Password		入力	ノードマネージャでSSLを使用する場合、認証に使用されるトラストストアファイルのパスワードを指定します。

確認再設定

● SSHタイプのノード・マネージャーの設定

[Nodes]画面で「Ssh」を選択します。設定するノード名、ホスト情報を入力して必要な項目を設定します。SSHタイプのノード・マネージャーの設定方法については、「[3.3.1. SSHタイプのノード設定](#)」を参照してください。

● Javaタイプのノード・マネージャーの設定

[Nodes]画面で「Java」を選択します。設定するノード名、ホスト情報を入力して必要な項目を設定します。Javaタイプのノード・マネージャーの設定方法については、「[2.4.1. Javaタイプのノード設定](#)」を参照してください。

1.3. 共通機能

以下は、2つのタイプのノード・マネージャーで共通して提供する機能です。

- リモート・マシンのサーバーの起動

ノード・マネージャーを利用してドメインに属しているサーバーを起動することができます。

ノード・マネージャーを使用しない場合は、サーバーを起動しようとするリモート・マシンに接続してスクリプトを利用してサーバーを起動します。しかし、ノード・マネージャーを使用すると、リモート・マシンに存在するサーバーがDASで簡単なコマンドを使って起動できます。DASが起動されている状態で、ドメインに属するすべてのMSを起動することができ、クラスターを起動することもできます。

ノード・マネージャーを使用してサーバーを起動するには、以下の条件を満たす必要があります。

- サーバーを起動しようとするノード情報がnodes.xmlに登録されている必要があります。
- どのノードで起動するのかが情報が、起動しようとするサーバーに設定されている必要があります。
- ノード・マネージャーに接続可能な状態である必要があります。

Javaタイプのノード・マネージャーを使用する場合は、ノード・マネージャーが起動状態である必要があります。一方、SSHタイプのノード・マネージャーを使用する場合は、OSにSSHポートが開いている必要があります。

- Javaタイプのノード・マネージャーを使用する場合、jeusnm.xmlにノード・マネージャーが設定されている必要があります。

- 異常なサーバーの再起動

Javaタイプのノード・マネージャーでは、モニタリングしているサーバーが異常終了した場合、プロセスを再起動します。SSHタイプのノード・マネージャーの場合は、ドメインのSCFによりサーバーが異常(FAILED)状態になると、DASではサーバーが属しているノード・マネージャーにサーバーの再起動コマンドを送り、サーバーが再起動できるようにします。

この動作は、該当するサーバーにノードが必ず設定されており、Javaタイプのノード・マネージャーの場合は、ノード・マネージャーが起動状態である必要があります。

参考

Javaタイプのノード・マネージャーとSshタイプのノード・マネージャーで提供する機能については、「[第2章 Javaタイプのノード・マネージャー](#)」と「[第3章 SSHタイプのノード・マネージャー](#)」を参照してください。

1.4. RQSプロセス管理機能

ノード・マネージャーは、サーバー・プロセス以外にも他のプロセスを管理できる機能があります。現在RQSプロセスの状態をチェックし、異常終了したと判断した場合、プロセスを再開始する機能を提供します。

- 当該ノードに設定されたRQSプロセスの実行・管理

ノード・マネージャーは設定されたRQSの情報でRQSを起動することができます。また設定されたポートを介してRQSプロセスの状態を継続してチェックすることができ、もし異常終了したと判断された場合は、RQSプロセスを再開始するようにして持続的なサービスを可能にします。

ノード・マネージャーでRQSプロセスを起動するには、以下の条件が満たされる必要があります。

- RQSプロセスの情報がjeusnm.xmlファイルに登録されている必要があります。
- RQSプロセスの起動に必要な設定ファイルが正しい場所に存在し、ノード・マネージャーに設定された設定値（パス、ポートなど）が正しく設定されている必要があります。設定ファイルの場所は、RQSプロセスがRQSDIR環境変数をベースに推定するので、環境変数の設定も重要です。
- ノード・マネージャーに接続可能な状態である必要があります。

- 異常のプロセスの再起動

RQSプロセスの管理に関するノード・マネージャーの主な役割は、RQSプロセスが異常終了したときに、それをチェックして再開始することです。そのために、指定されたポートにアクセスして相互にメッセージをやり取りすることになります。

RQSが異常終了したと判断すると、ノード・マネージャーは設定値をベースにRQSを再起動します。RQSが起動した共有リソースを整理するためには、-rオプションを使ってRQSを起動します。

参考

RQSプロセスの管理については、「[第4章 RQSプロセス管理](#)」を参照してください。

第2章 Javaタイプのノード・マネージャー

本章では、Javaタイプのノード・マネージャーを使用するための設定方法と動作方式について説明します。

2.1. 概要

Javaタイプのノード・マネージャーは、SSHタイプのノード・マネージャーに比べてサーバーの異常終了を迅速に検知し、再起動できるメリットがあります。また、OSがWindowsの場合、SSHタイプのノード・マネージャーは使用できませんが、Javaタイプのノード・マネージャーはOSを問わず使用できます。しかし、別のマシンにJEUSをインストールすることができないデメリットがあります。

以下では、Javaタイプのノード・マネージャーを使用するための方法と基本知識について記述します。

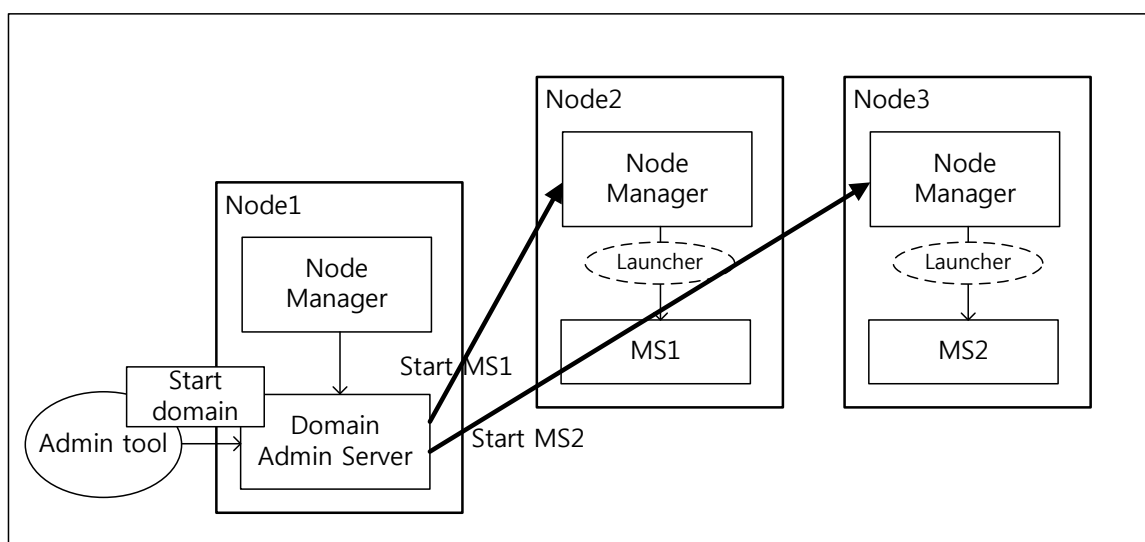
2.2. 機能別の動作方式

本節では、Javaタイプ・ノード・マネージャーの各機能の動作方式について説明します。

2.2.1. リモート・マシンに存在するMS(ManagedServer)の起動

ノード・マネージャーを使用してドメイン内のリモート・マシンに存在するサーバーを起動することができます。その手順は以下のとおりです。

【図 2.1】 ノード・マネージャーでリモート・マシンのサーバー起動



1. ノード・マネージャーでサーバーのlockファイルを作成してFILE LOCKを得ます。
2. サーバーが正常に起動したら、pidファイル、stateファイルを作成し、PID(プロセスID)とサーバーの状態を記録します。
3. <serverName>.propertiesにusername、password、dasurlなどの情報をアップデートします。
4. サーバーでは、NodeManagerServiceを開始すると同時にaddressファイルを作成し、自身のホスト情報を記録します。

参考

サーバーを再起動する場合も同様な手順で行われます。

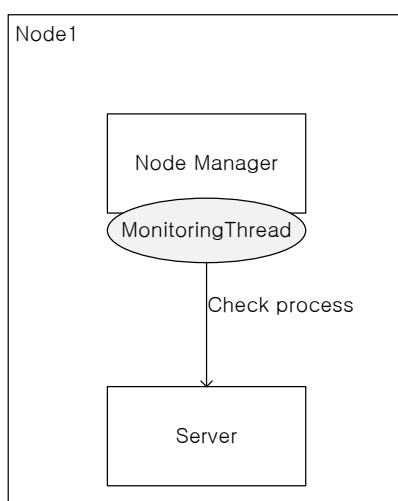
2.2.2. サーバー・モニタリング

Javaタイプのノード・マネージャーは、サーバー・プロセスの状態を周期的にモニタリングしてstateファイルにアップデートし、異常状況が発生した場合はサーバー・プロセスを再起動します。

ノード・マネージャーは、ノードに存在するすべてのサーバーをモニタリングするのではなく、ノード・マネージャーを介して起動したサーバー・プロセスのみをモニタリングします。したがって、サーバーの異常終了を検知したら直ちに再起動したい場合は、ノード・マネージャーを使用して起動する必要があります。ノード・マネージャーが再起動された場合にも、自身が起動したサーバーに対するモニタリングは継続されます。

以下は、Javaタイプのノード・マネージャーがサーバーをモニタリングするプロセスを簡単に表した図です。

[図 2.2] Javaタイプのノード・マネージャーでのサーバー・モニタリング



以下は、ノード・マネージャーが再起動された場合、自身が起動したサーバーを見つけ、継続してモニタリングする手順についての説明です。

1. ノード・マネージャーが起動されるとき、以下のパスでノード・マネージャーが管理しているドメイン名を照会します。

```
JEUS_HOME/domains
```

2. 各ドメインの以下のパスのサーバー・ディレクトリーを利用して、ノード・マネージャーが管理しているサーバー名を照会します。

```
JEUS_HOME/domains/<domain-name>/servers
```

3. サーバー・ディレクトリーで<serverName>.address、<serverName>.lck、<serverName>.state、<serverName>.pidファイルの存在有無を確認します。すべてのファイルが存在する場合、サーバーは運用中の状態になります。
4. lockファイルを使用してFILE LOCKを取得します。
5. stateファイルに記録されたサーバーの状態を確認し、サービス可能な状態なのかを判断します。
 - サービスできる状態 : RUNNING、STANDBY、SUSPENDED、RESUMING、SUSPENDING、STARTING
 - サービスできない状態 : SHUTDOWN、SUNTTING_DOWN
6. pidファイルに記録されたサーバー・プロセスIDを使って、プロセスが運用中であるかどうかを確認します。サーバーのダウンが確認されたら、ノード・マネージャーは当該サーバーを再起動します。
7. addressファイルに記録されたサーバーのホスト情報をベースにして、ソケット接続を行ってサーバー・プロセスが正常に運用されているかを確認します。サーバーのダウンが確認されたら、ノード・マネージャーは当該サーバーを再起動します。
8. サーバーがサービス可能な状態であると判断された場合、周期的にサーバーの状態を確認しながら、モニタリングします。

参考

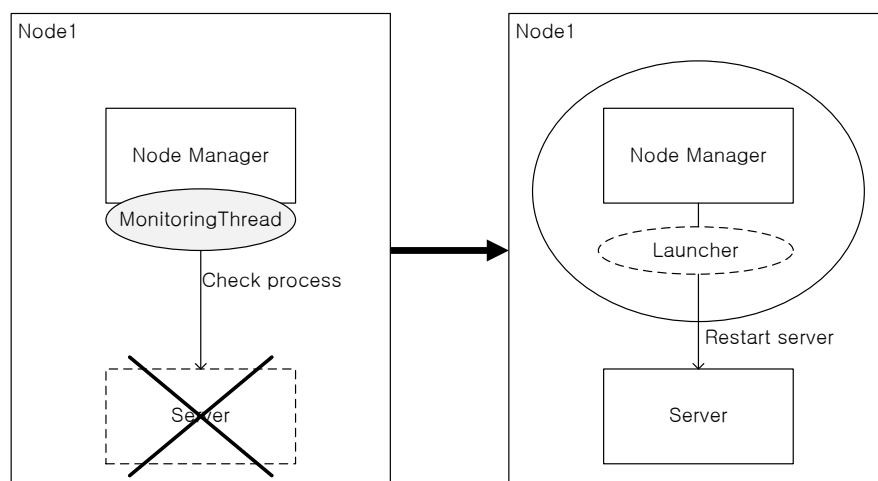
stateファイルに記録されるサーバー状態のうちSHUTDOWN_NEED_TO_RESTART状態は、ノード・マネージャーが再起動されるとき、以前のノードでサービスしていたサーバーを再起動するために使用されるstateです。主に、Windowsサービスのようにノード・マネージャーをOSのサービスとして登録した場合、サーバーがシャットダウンしてから再起動されるとき、以前サービス中であったサーバーを再起動してサービスできるようにするために使用されます。

2.2.3. 異常なサーバーの再起動

Javaタイプのノード・マネージャーは、サーバー・プロセスが異常終了された場合に再起動を行います。サーバーが定義した異常状態になった場合にもサーバーを再起動します。

サーバーではメモリー使用率(Memory Ratio)を設定して、指定した割合以上にヒープ・メモリーを使用した場合はプロセスを終了します。ノード・マネージャーは、サーバーが異常状況によって終了されるのを検知して、サーバーを再起動します。

【図 2.3】 Javaタイプのノード・マネージャーのサーバー再起動



2.2.4. ローリング・パッチ

Javaタイプのノード・マネージャーを利用してドメインにパッチを適用することができます。コンソール・ツールを使ってリモート・マシンに存在するサーバーにパッチ・ファイルを転送または削除したり、パッチを適用するためにサーバーを再起動したりします。

DASからドメインに属しているサーバーが存在するリモート・マシンのノード・マネージャーに接続してパッチ・ファイルを転送、削除、適用することができます。

- パッチ・ファイルの転送

```
domain1.adminServer>apply-patch
```

- パッチ・ファイルの削除

```
domain1.adminServer>remove-patch
```

- パッチ・ファイルの適用

-rollingオプションを使って各サーバーにパッチ・ファイルを適用することができます。その際、ノード別に順次実行し、サービスが二重化されている場合でも正常にサービスできるようにします。

ただし、同じノード内に複数のサーバーを運用している場合は、ノード全体を1つずつ順次に終了し、パッチ・ファイルを転送または削除した後、再びサーバーを1つずつ順次起動する必要があります。

```
domain1.adminServer>apply-patch -rolling

domain1.adminServer>remove-patch -rolling
```

- パッチ適用に失敗した場合にロールバックする機能

-actionオプションを使ってパッチ適用に失敗した場合の動作について定義することができます。

```
domain1.adminServer>apply-patch -rolling -action CONTINUE|STOP|ROLLBACK
```

以下のCONTINUE、STOP、ROLLBACKオプションから1つが設定できます。

オプション	説明
CONTINUE	Pパッチの適用に失敗したノードがあっても無視して続けます
STOP	パッチの適用に失敗したノードがあった場合、パッチの適用を中断してリターンします
ROLLBACK	パッチの適用に失敗したノードがあるとパッチの適用を中断し、パッチの適用に成功したノードに対してもロールバックします。 パッチを削除する際、パッチ・ファイルがすでに削除されたノードが存在する可能性があるため、同オプションを設定してもSTOPオプションを設定した場合と同様、それ以上のパッチ削除を中断し、動作を止めます

2.2.5. DASとの接続

ノード・マネージャーを使用せずに起動されたサーバーはノード・マネージャーの管理が受けられないため、モニタリング機能やパッチ機能を提供しません。

しかし、DASは例外的にノード・マネージャーを介さずに起動されてもプロセス・モニタリング機能やパッチの転送が可能です。DASは、起動が完了されるとノード・マネージャーに接続して自身の情報を転送するため、ノード・マネージャーではこの情報をベースにしてDASをモニタリングできます。

ローリング・パッチ機能は、DASとノード・マネージャーが接続されている状態でのみ使用できます。DASがノード・マネージャーを介して起動されていなくても、DASとノード・マネージャーが接続された状態なので、自身のノードのみならず、別のノードにもパッチの転送および適用が可能です。

参考

ノード・マネージャーを介さずに起動されたサーバーはモニタリングしません。Javaタイプのノード・マネージャーを使用すると、ノード・マネージャーを介して起動されたサーバーは、サーバーがFAILED状態になる前にノード・マネージャーがサーバーの異常終了を検知して再起動することができます。

2.3. 環境設定

Javaタイプのノード・マネージャーを使用するために必要な環境を設定します。

2.3.1. 設定ファイル

Javaタイプのノード・マネージャーを使用するには、以下の設定ファイルに必要な事項を設定する必要があります。Javaタイプのノード・マネージャーは、nodes.xmlにノードが定義されること以外にも別途の設定が必要です。

- nodes.xml
- jeusnm.xml
- <serverName>.properties

nodes.xml

使用するノード・マネージャーのタイプをnodes.xmlファイルに設定します。ファイルは以下のパスに格納されます。

```
JEUS_HOME/domains
```

nodes.xmlはDASで参照するノードの設定です。ノードの物理的なアドレスを定義し、ドメイン設定では各MSが運用されるノード情報を設定します。このファイルは、1つのドメインに限らず、マシンにインストールされたJEUSに存在する複数のドメインで共有されます。各ドメインのDASでは、node.xmlに設定されたノード情報によって各ノードのノード・マネージャーに接続して必要なコマンドを実行します。

WebAdminまたはコンソールを使用してノードを追加、削除することができます。ノードの追加、削除についての詳細方法は、[「2.4. ノードの設定および削除」](#)を参照してください。

参考

Javaタイプのノード・マネージャーは、nodes.xmlに設定されたマシン情報と、jeusnm.xmlに設定された「host」、「port」情報が一致するように注意して設定します。

jeusnm.xml

jeusnm.xmlは、マシンにおけるノード・マネージャーの動作方式を設定する設定ファイルです。以下のパスに存在します。

```
JEUS_HOME/nodemanager
```

jeusnm.xmlに設定できる項目は以下のとおりです。jeusnm.xmlファイルは直接修正します。

項目	説明
useNodeManager	Javaタイプのノード・マネージャーの使用有無を設定します(デフォルト値: true)
host	ノード・マネージャーのリスン・アドレスを設定します(デフォルト値: localhost)
port	ノード・マネージャーのリスン・ポートを設定します(デフォルト値: 7730)
serverMonitoringPeriod	ノード・マネージャーでサーバー・プロセスの状態をモニタリングする周期を設定します(デフォルト値: 500、単位: ms)
serverAutoRestart	サーバー・プロセスが異常終了された場合、自動的にサーバーを再起動するか否かを設定します(デフォルト値: true)
serverRestartTryCount	サーバーを再起動する回数を設定します。(デフォルト値: 5) 「 serverRestartDurationTime 」に設定した周期内に、指定した回数以上のサーバー再起動が行われた場合、サーバー・プロセスに問題があると判断してサーバーの再起動を中断します
serverRestartDurationTime	サーバーを再起動する周期を設定します。(デフォルト値: 120000、単位: ms) 設定した周期内に「 serverRestartTryCount 」に指定した回数以上のサーバー再起動が行われる場合、サーバー・プロセスに問題があると判断してサーバーの再起動を中断します
serverRetryRestart	サーバー再起動に失敗した場合、再実行するか否かを指定します。この値をtrueに設定すると、サーバーの再起動に成功するまで継続してトライします(デフォルト値: false)
useSSLListener	ノード・マネージャーのSSL使用有無を設定します(デフォルト値: false)
keystoreFile	SSLを使用する際、認証に使われるキーストア・ファイルのパスを設定します (デフォルト値: NODEMANAGER_HOME/keystore)
keystorePass	SSLを使用する際、認証に使われるキーストア・ファイルのパスワードを設定します (デフォルト値: jeuskeypass)
truststoreFile	SSLを使用する際、認証に使われるトラストストア・ファイルのパスを設定します (デフォルト値: NODEMANAGER_HOME/truststore)
truststorePass	SSLを使用する際、認証に使われるトラストストア・ファイルのパスワードを設定します (デフォルト値: jeustruststorepass)

項目	説明
logFileName	ノード・マネージャーのログファイルのパスを設定します。ファイルに絶対パスを与えた場合はそのパスにログファイルを作成し、相対パスを設定した場合はデフォルトのログ・ディレクトリーに設定した名前でログファイルを作成します (デフォルト値: JEUS_HOME/nodemanager/logs/JeusNodeManager.log)
standbyPort	ノード・マネージャーのスタンバイ・プロセスを起動して、ノード・マネージャーが異常終了したときに自動でスタンバイ・プロセスが代替するようにできます。使用するポートを設定することで機能を有効にすることができます。詳細については、「 5.3. 冗長化されたノード・マネージャーの使用方法 」の「 スタンバイ・ノード・マネージャー 」を参照してください
processList	RQSプロセスの情報をリスト形式で設定します。設定方法については、「 4.3.1. 設定ファイル 」を参照してください

<serverName>.properties

<serverName>.propertiesファイルは、ノード・マネージャーでサーバーを起動する際に必要な情報のキャッシュ・ファイルです。ノード・マネージャーでサーバーを起動した後、以下のパスにファイルを作成します。

```
SERVER_HOME/nodemanager
```

ノード・マネージャーは、サーバーを起動するとき必要なオプションを記録するので、次回にオプションを設定しなくても同ファイルにキャッシングされている情報を利用してサーバーを起動させることができます。この情報は、コンソール・ツールを使ってノード・マネージャーに直接接続してサーバーを起動する場合のみ使用されます。DASを介してMSを起動する場合は当該情報を使用しません。

キャッシュされる情報は以下のとおりです。

項目	説明
dasurl	DASのホスト情報です。MSの場合のみdasurlをキャッシュします
username	サーバーを起動する際に必要なサーバーのアカウントです
password	サーバーを起動する際に必要なサーバーのパスワードです
sslArguments	SSLを使用するように設定されたサーバーの場合、ノード・マネージャーからサーバーに接続する際のSSL認証が必要なSSL properties情報を設定します

2.3.2. 必須ファイル

以下では、設定ファイル以外にノード・マネージャーがサーバーを起動し、モニタリングするために必要なファイルについて説明します。各ファイルは次のパスに存在します。

- **<serverName>.lck**

サーバー・プロセスを起動する際、サーバーが数回起動されるのを防ぐためにノード・マネージャーが生成するファイルです。同ファイルは、ノード・マネージャーでサーバー・プロセスを開始する前に生成され、サーバー・プロセスが正常に終了すると削除されます。ノード・マネージャーが再起動されたとき、同ファイルの有無によってサーバーを継続してモニタリングするか否かが決まります。

- **<serverName>.pid**

サーバー・プロセスが正常に起動された後、ノード・マネージャーが生成するサーバーのプロセスIDを記録するファイルです。同ファイルは、ノード・マネージャーでサーバー・プロセスを正常に起動させた後に生成され、サーバー・プロセスが正常に終了されると削除されます。ノード・マネージャーが再起動されたとき、同ファイルの有無によってサーバーを継続してモニタリングするか否かが決まります。また、ファイルに記録されたPIDに該当するプロセスの存在有無からサーバーの異常終了を判断し、ノード・マネージャーが再起動された後にもサーバーを再起動して継続してモニタリングすることができます。

- **<serverName>.state**

サーバー・プロセスの起動が正常に完了した後、ノード・マネージャーで生成するサーバーの状態を記録するファイルです。同ファイルは、ノード・マネージャーでサーバー・プロセスを正常に起動させた後に生成され、サーバー・プロセスが正常に終了されると削除されます。ノード・マネージャーでは周期的にサーバーの状態をチェックし、同ファイルにサーバーの状態をアップデートします。ノード・マネージャーが再起動されたとき、同ファイルの有無によってサーバーを継続してモニタリングするか否かが決まります。なお、ファイルに記録されたサーバーの状態を確認し、サーバーの異常終了を判断してノード・マネージャーが再起動された後にもサーバーを再起動して継続してモニタリングすることができます。

- **<serverName>.address**

サーバーが起動するときに生成するファイルです。サーバーのListen AddressとListen Portが記録されます。同ファイルはサーバーが起動されるときに生成され、正常に終了されると削除されます。ノード・マネージャーが再起動されたとき、同ファイルの有無によってサーバーを継続してモニタリングするか否かが決まります。また、ファイルに記録されたホスト情報を利用してサーバーを継続してモニタリングすることができます。ノード・マネージャーが再起動されたときは、サーバーのMBeanを使用してサーバーの状態を周期的にモニタリングし、異常終了の有無をチェックします。

2.4. ノードの設定および削除

Javaタイプのノードは、WebAdminとコンソール・ツールを使用して設定および削除することができます。

2.4.1. Javaタイプのノード設定

本節では、WebAdminとコンソール・ツールを使用してJavaタイプのノードを設定する方法について説明します。

参考

WebAdminまたはコンソール・ツールを使用してサーバーを起動するには、サーバーにノード名が設定されている必要があります。

WebAdminの使用

以下は、WebAdminを使用してJavaタイプのノードを設定する方法です。

1. ノードに関連するすべての編集および動作は、**[ノード設定]**メニューを利用して実行できます。WebAdminのメイン画面にて**[ノード設定]**をクリックします。
2. **[Nodes]**画面には追加されたノード・リストが表示されます。ノード・リストには、ユーザーがノードを追加しなくてもJEUSをインストールする際に基本的に生成されたノードが照会されます。ノードを追加するため、**[Add]**ボタンをクリックします。

参考

ノード設定は、**[Lock & Edit]**ボタンを使用した設定変更モードへの切り替えや、**[Apply Changes]**を使用した変更の反映のような動作は不要です。

3. **[Node]**画面でノード名、ホスト情報を入力します。ノード・タイプに「**Java**」を選択して関連項目を入力した後、**[確認]**ボタンをクリックします。
4. **[Nodes]**画面の上部にノードが正常に追加されたとのメッセージと、ノード・リストに追加されたノード情報が表示されます。

コンソール・ツールの使用

以下は、コンソール・ツールを使用してノードを設定する方法です。

1. **add-java-node**コマンドを使って「node1」というノードを追加します。

```
[DAS]domain1.adminServer>add-java-node node1 -host 192.168.34.65 -port 7730
The node [node1] was successfully added.
```

2. **show-node**コマンドを使って追加されたノードの設定を照会します。Javaタイプの「node1」が照会されることを確認できます。

```
[DAS]domain1.adminServer>show-node node1
=====
+-----+-----+
|                Property                |      Value      |
+-----+-----+
| Node Name          | node            | |
| Host               | 192.168.34.65  |
| Mapped Servers     |                 |
| Node Type          | JAVA           |
| NodeManager Port   |                | 7730             |
| Use SSL            | false          |
+-----+-----+
=====
```

また、**list-nodes**コマンドを使ってドメインのノードを照会することができます。

```
[DAS]domain1.adminServer>list-nodes
=====
+-----+-----+-----+-----+
| Node name | Type | Under control | JEUS version |
+-----+-----+-----+-----+
| node1     | JAVA | N             | -            |
+-----+-----+-----+-----+
=====
```

3. **modify-java-node**コマンドまたは**modify-node**コマンドを使ってノードを変更することができます。

```
[DAS]domain1.adminServer>modify-java-node node1 -port 7731
The node [node1] was modified successfully. Check the results using "show-node"
[DAS]domain1.adminServer>modify-node node1 -port 7732
The node [node1] was modified successfully. Check the results using "show-node"
```

4. 変更されたノード情報を確認するには、**show-node**コマンドを使用して該当するノードの設定を照会します。

```
[DAS]domain1.adminServer>show-node node1
=====
+-----+-----+
|                Property                |      Value      |
+-----+-----+
| Node Name          | node            | |
| Host               | 192.168.34.65  |
| Mapped Servers     |                 |
| Node Type          | JAVA           |
| NodeManager Port   |                | 7732             |
+-----+-----+
```

Use SSL	false	
+-----+-----+		
=====		

参考

add-java-nodeコマンドとlist-nodesコマンドの詳しい使用方法については、『*JEUS リファレンスガイド*』の「4.2.15. ノード管理関連コマンド」を参照してください。

2.4.2. Javaタイプのノード削除

本節では、WebAdminとコンソール・ツールを使用してJavaタイプのノードを削除する方法について説明します。

WebAdminの使用

以下は、WebAdminを使用してノードを削除する方法です。

1. WebAdminのメイン画面で**[ノード設定]**を選択します。**[Nodes]**画面に照会されたノード・リストから削除するノードの**[delete]**ボタンをクリックします。
2. ノード削除についてのメッセージが表示され、**[確認]**ボタンをクリックすると当該ノードは削除されます。
3. **[Nodes]**画面の上部にノードが正常に削除されたとのメッセージが表示され、ノード・リストから削除したノード情報が照会されないことが確認できます。

コンソール・ツールの使用

以下は、コンソール・ツールを使ってノードを削除する方法です。

1. remove-nodeコマンドを使って「node1」というノードを削除します。

```
[DAS]domain1.adminServer>remove-node node1
The node [node1] was successfully removed.
```

2. 追加されたノードを確認するには、list-nodesコマンドを使ってノード・リストを照会します。ノード・リストにJavaタイプの「node1」が照会されないことが確認できます。

```
[DAS]domain1.adminServer>list-nodes
=====
+-----+-----+-----+-----+
| Node name | Type | Under control | JEUS version |
```

```
+-----+-----+-----+-----+-----+
(No data available)
=====
```

参考

remove-nodeとlist-nodeコマンドの詳しい使用方法については、『*JEUS リファレンスガイド*』の「4.2.15. ノード管理関連コマンド」を参照してください。

2.5. ノード・マネージャーの開始および終了

Javaタイプのノード・マネージャーは、SSHタイプのノード・マネージャーとは違って、別途の開始および終了プロセスが必要です。

Javaタイプのノード・マネージャーの開始

Javaタイプのノード・マネージャーはスクリプトを利用して実行させます。JEUS_HOME/bin下位に存在する**startNodeManager**を実行すると、以下のようにノード・マネージャーがDASまたはコンソール・ツールからの指示を受け、サーバーを制御する準備をします。

```
JEUS_HOME/bin$ startNodeManager
*****
- JEUS Home           : /home/jeus/jeus8
- Added Java Option  :
- Java Vendor        : Sun
*****
...
[2016.07.28 13:55:32][2] [nodemanager-1] [NodeManager-0101] The node manager is
starting.
[2016.07.28 13:55:32][2] [nodemanager-1] [NodeManager-0102] Initializing the node
manager configuration.
[2016.07.28 13:55:32][2] [nodemanager-1] [NodeManager-0108] Beginning to listen:
localhost/127.0.0.1:7730.
[2016.07.28 13:55:32][2] [nodemanager-10] [NodeManager-0109] Processing the
request.....
```

ノード・マネージャーがサーバーをモニタリングする最中に終了され、再起動された場合、以前モニタリングしていたサーバーを引き続きモニタリングするための作業を行います。その際、以下のようなログが発生します。

```
JEUS_HOME/bin$ startNodeManager
*****
- JEUS Home           : /home/jeus/jeus8
- Added Java Option  :
- Java Vendor         : Sun
*****
...
[2016.07.28 13:59:16][2] [nodemanager-1] [NodeManager-0101] The node manager is
starting.
[2016.07.28 13:59:16][2] [nodemanager-1] [NodeManager-0102] Initializing the
node manager configuration.
[2016.07.28 13:59:16][2] [nodemanager-1] [NodeManager-0108] Beginning to listen:
localhost/127.0.0.1:7730.
[2016.07.28 13:59:16][2] [nodemanager-1] [NodeManager-0115] Domain=[domain1],
Server=[adminServer]
[2016.07.28 13:59:16][2] [nodemanager-10] [NodeManager-0109] Processing the
request.....
[2016.07.28 13:59:16][2] [nodemanager-1] [NodeManager-0115] Domain=[domain1],
Server=[server1]
[2016.07.28 13:59:16][2] [nodemanager-11] [NodeManager-0137] Beginning to
monitor the server[adminServer] in the domain[domain1].
[2016.07.28 13:59:16][2] [nodemanager-12] [NodeManager-0137] Beginning to
monitor the server[server1] in the domain[domain1].
[2016.07.28 13:59:16][2] [nodemanager-11] [NodeManager-0145] The process is
alive(Server=adminServer, Process ID=4856).
[2016.07.28 13:59:17][2] [nodemanager-12] [NodeManager-0145] The process is
alive(Server=server1, Process ID=5376).
```

参考

ノード・マネージャーを起動するとき、DASも一緒に起動することができます。ドメインを構成して最初にノード・マネージャーを起動するとき、DASの起動オプションを設定した場合は、該当するノードにDASを起動させます。

startNodeManager -hにより、使用できるオプションを確認することができます。

Javaタイプのノード・マネージャーの終了

Javaタイプのノード・マネージャーを終了するために、スクリプトまたはコンソール・ツールを使用することができます。

- スクリプトの使用

起動するときと同様、スクリプトを使用してノード・マネージャーを終了することができます。**stopNodeManager**を実行すると、Javaタイプのノード・マネージャーが終了します。

```
JEUS_HOME/bin$ stopNodeManager -host localhost -port 7730
*****
- Usage : stopNodeManager -host host -port port
*****
Succeed to stop the node manager.
```

ノード・マネージャーを終了するとき、-propertiesオプションを使用してjeusnm.xmlファイルのパスも設定することができます。ノード・マネージャーの設定ファイルからホストとポート情報を読み込んでノード・マネージャーを終了します。

```
JEUS_HOME/bin$ stopNodeManager -properties JEUS_HOME/nodemanager/jeusnm.xml
*****
- Usage : stopNodeManager -host host -port port
*****
Succeed to stop the node manager.
```

- コンソール・ツールの使用

コンソール・ツールを使用して**nm-stop**コマンドを実行するとノード・マネージャーが終了します。nm-stopコマンドの詳しい使用方法については、『*JEUS リファレンスガイド*』の「4.2.16.6. stop-nodemanager」を参照してください。

```
JEUS_HOME/bin$ jeusadmin
JEUS8 Administration Tool
To view help, use the 'help' command.
offline>nm-stop -host localhost -port 7730
Succeed to stop the node manager.
```

2.6. Javaタイプのノード・マネージャーを使用したサーバーの制御

Javaタイプのノード・マネージャーを使ってサーバーを制御することができます。本節では、WebAdminとコンソール・ツールを使用したサーバーの制御方法について説明します。

WebAdminの使用

WebAdminを使用してサーバーを起動すると、DASがサーバーを起動するマシンのノード・マネージャーに接続してサーバー起動コマンドを送信し、ノード・マネージャーはサーバーを当該マシンで起動させます。

1. WebAdminのメイン画面にて**[Servers]**をクリックすると、サーバー・リストが表示されます。

参考

サーバーを起動する方法は、[Servers]メニューの以外にも、[Monitoring] > [Servers]メニューを使用することができます。当該メニューを使用する方法については、「[第3章 SSHタイプのノード・マネージャー](#)」を参照してください。

2. 起動するサーバーの[start]ボタンをクリックすると、[Start Server]画面でサーバー起動に関するオプションを設定することができます。必要な事項を設定した後、[確認]ボタンをクリックしてサーバーを起動します。
3. [Servers]画面の上部にサーバーが正常に起動されたとのメッセージと、サーバー・リストのサーバー状態がRUNNINGに変更されたことが確認できます。

コンソール・ツールの使用

ノード・マネージャーがサーバーを起動/終了させ、サーバーの状態を確認するなどの制御は、コンソール・ツールを使用することもできます。

コンソール・ツールを使ってサーバーを制御するには、コンソール・ツールとノード・マネージャーが接続されている必要があります。必要な作業が完了すると接続を終了します。

- ノード・マネージャーに接続

nm-connectコマンドを使ってノード・マネージャーに接続します。

```
JEUS_HOME/bin$ jeusadmin
JEUS8 Administration Tool
To view help, use the 'help' command.
offline>nm-connect -host localhost -port 7730 -domain domain1
The connection to the node manager domain1 has been established.
[NodeManager]domain1>
```

- ノード・マネージャーを使用したサーバーの起動

nm-start-serverコマンドを使ってサーバーを起動します。

```
[NodeManager]domain1>nm-start-server -server adminServer -u jeus -p jeus
succeed to start server[adminServer].
RUNNING
[NodeManager]domain1>nm-start-server -server server1 -u jeus -p jeus
-dasurl 192.168.0.4:9736
succeed to start server[server1].
RUNNING
```


以下のようにnm-start-serverコマンドを使って、ノード・マネージャーへの接続を一度に実行することもできます。

```
JEUS_HOME/bin$ jeusadmin
offline>nm-start-server -host localhost -port 7730 -domain domain1
-server adminServer -u jeus -p jeus
succeed to start server[adminServer].
RUNNING
```

- スクリプトを使用してノード・マネージャーの制御を受けるDASを起動

DASに限ってはノード・マネージャーを起動しなくても、ノード・マネージャーと接続し、ノード・マネージャーの制御を受けることができます。

startDomainAdminServerNMスクリプトを使用すると、ノード・マネージャーに接続し、DAS起動コマンドを実行してノード・マネージャーがDASを起動させます。コンソール・ツールを使用してノード・マネージャーに接続し、サーバーを起動するとの2つのコマンドを1つのスクリプトで提供します。

参考

startDomainAdminServerスクリプトを利用してDASを起動した場合でもノード・マネージャーの管理を受けることができます。DASは起動された後、ノード・マネージャーへの接続をトライして自身の情報を登録します。ノード・マネージャーにDASの情報が登録されると、ノード・マネージャーでDASの異常状態を把握し、再起動するなどの制御が可能となります。

```
JEUS_HOME/bin$ startDomainAdminServerNM -host 192.168.0.26 -port 7730
-domain domain1 -server adminServer -u jeus -p jeus
*****

- Usage : startDomainAdminServerNM -host host -port port -domain domain -server
server1 -u username -p password
*****

succeed to start server[adminServer].
RUNNING
```

- ノード・マネージャーを使用したサーバー状態の確認

nm-state-serverコマンドを使ってサーバーの状態を確認することができます。

```
[NodeManager]domain1>nm-state-server -server adminServer -u jeus -p jeus
server[adminServer] : RUNNING
[NodeManager]domain1>nm-state-server -server server1 -u jeus -p jeus
server[server1] : RUNNING
```

- ノード・マネージャーを使用したサーバーの終了

nm-stop-serverコマンドを使ってサーバーを終了することができます。

```
[NodeManager]domain1>nm-stop-server -server server1 -u jeus -p jeus
succeed to stop server[server1].
[NodeManager]domain1>nm-stop-server -server adminServer -u jeus -p jeus
succeed to stop server[adminServer].
```

- ノード・マネージャーとの接続終了

nm-disconnectコマンドを使ってノード・マネージャーとの接続を終了します。

```
[NodeManager]domain1>nm-disconnect
disconnect to node manager.
offline>
```

- DAS接続モードでのサーバー起動

start-serverコマンドを使ってMSを起動する際、ノード・マネージャーを介してサーバーを起動します。

```
[DAS]domain1.adminServer>start-server server1
The server [server1] was successfully started.
```

参考

コンソール・ツールでJavaタイプのノード・マネージャーに接続・終了するコマンドや、Javaタイプのノード・マネージャーを利用したサーバー制御コマンドの詳細な使用方法については、『*JEUS リファレンスガイド*』の「4.2.15. ノード管理関連コマンド」を参照してください。

2.7. ログ・ファイル

ノード・マネージャーのログは、JEUS_HOME/nodemanager/logsの下位のJeusNodeManager.logというファイルに記録されます。ノード・マネージャーでサーバーを起動、モニタリングするときに発生するログ・メッセージを記録します。また、サーバーが起動される際、サーバーで発生する起動ログ・メッセージも記録されます。

ノード・マネージャーのログ・レベルを設定するには、ノード・マネージャーを起動するスクリプトに以下のように設定します。

```
-Djeus.nodemanager.log.level=FINEST
```

ログとログ・レベルについての詳細については、『*JEUS サーバガイド*』の「第8章 ログイン」を参照してください。

第3章 SSHタイプのノード・マネージャー

本章では、SSHタイプのノード・マネージャーを使用するための設定方法について説明します。

3.1. 概要

SSHタイプのノード・マネージャーはUNIX環境の運用システムで使えるノード・マネージャー・タイプとして、ノードにSSH(Secure Shell)コマンドを実行して動作します。ノードにJEUSサーバーを追加するとき、簡単にインストールできる機能を提供し、DASマシンでSSHサーバーを実行してローカル・ノードを構成することもできます。ただし、Windows環境の場合はSSHタイプのノード・マネージャーはサポートしません。

参考

本章では、リモート設定とローカル(DASマシン)設定の混乱を避けるため、リモート・ノード、リモート・サーバーを基準にして説明します。

3.2. 環境設定

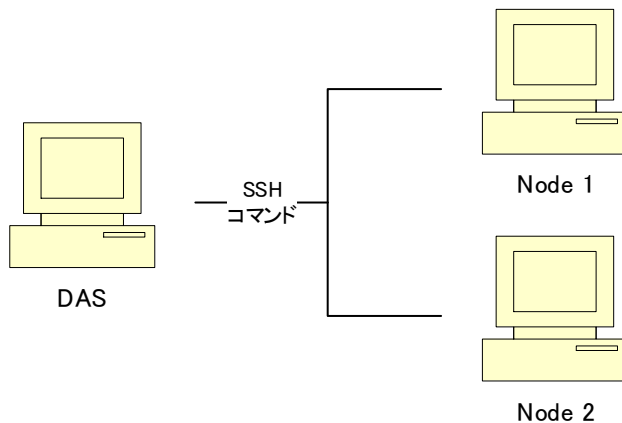
SSHタイプのノード・マネージャーを使用するためには、いくつかの環境設定が必要です。

3.2.1. SSHの設定

UNIX環境の運用システムでSSHタイプのノード・マネージャー機能を使用するには、DASマシンからノードにSsh接続が可能である必要があります。すなわち、DASマシンはSSHクライアントになり、リモート・ノードはSSHサーバーとなります。

本節では、SSHサーバーの動作についての説明は省略します。SSH接続に必要な設定についてのみ説明します。

[図 3.1] リモート・ノードへのSSH接続



JEUSからリモート・ノードへのSSH接続は、個人キーを使った場合のみ許可します。個人キーを使用した接続のために、DASマシンでssh-keygenを利用して公開キーと個人キーを生成します。キーを生成する際、個人キーを暗号化するパスフレーズ(passphrase)はJEUSでサポートしないため、設定しません。

生成した個人キーと公開キーは、USER_HOME/.sshフォルダーに「id_rsa」、「id_rsa.pub」という名前のファイルで格納されます。公開キー「id_rsa.pub」の内容を、接続するリモート・ノードのUSER_HOME/.ssh/authorized_keysファイルに追加すると、個人キーを使用した接続が可能となります。

以下のようにsshコマンドを実行して、個人キーを使ったSSH接続の可能有無が確認できます。

```
ssh <ssh-user-name>@<remote-host-address>
```

以下は、sshコマンドの例です。

```
$ ssh jeus@192.168.23.129
```

sshコマンドを使ってリモート・サーバーを実行するには、ノードにJAVA_HOMEが環境変数で設定されている必要があります。SSHはコマンドを実行するとき、リモート・マシンのUSER_HOME/.bashrcファイルを利用して設定される環境変数を使用するため、.bashrcファイルを以下のように修正します。

JEUSでのSSHコマンドの実行は、基本的にnon-interactiveモードで動作します。non-interactiveモードでは、設定した環境変数が動作しない場合があるため、以下のとおりルートに環境変数を設定します。

```
# ~/.bashrc

export JAVA_HOME='/home/java/jdk/jdk1.7.0_80'
export PATH='/home/java/jdk/jdk1.7.0_80/bin':$PATH

# If not running interactively, don't do anything
[ -z "$PS1" ] && return

...
```

Kシェル(Korn shell)の場合は、USER_HOME/.kshrcに上記の内容を設定します。なお、USER_HOME/.ssh/environmentファイルを新規生成してENV=~/.kshrcを追加します。

```
$ cat ~/.ssh/environment
ENV=~/.kshrc
```

上記の設定が正常に動作するには、/etc/ssh/sshd_configファイルのPermitUserEnvironmentを「yes」に設定し、sshdを再起動します。

リモート・ノードにJAVA_HOME環境変数を設定した後、DASマシンで以下のコマンドを使って設定が正常に適用されたのかを確認します。

```
$ ssh jeus@192.168.23.129 java -version
java version "1.6.0_30"
Java(TM) SE Runtime Environment (build 1.6.0_30-b12)
Java HotSpot(TM) Server VM (build 20.5-b03, mixed mode)
```

上記のコマンドが正常に動作されたら、SSHを使用する準備が完了したことになります。

3.3. ノードの設定および削除

SSHタイプのノードは、WebAdminとコンソール・ツールを使用して設定および削除することができます。

3.3.1. SSHタイプのノード設定

SSH設定が完了したら、JEUSにノードの設定を適用します。

ノードには、基本的に以下の3つの基本情報が必要です。同情報と共にノードのタイプ別に必要な設定を行います。

- IDとして使用するノード名
- リモート・マシンのアドレス
- リモート・マシンのJEUSインストール・ディレクトリーの情報

SSHタイプのノード・マネージャーは、UNIX環境でのみサポート可能です。SSH接続のためにポート番号、ユーザー名、個人キー・ファイルのパス情報を設定します。本節では、WebAdminとコンソール・ツールを使用してJEUSにSSHタイプのノードを設定する方法について説明します。

WebAdminの使用

以下は、WebAdminを使用してSSHタイプのノードを設定する方法です。

1. ノードに関連するすべての編集および動作は、**[ノード設定]**メニューを利用して実行できます。WebAdminのメイン画面にて**[ノード設定]**をクリックします。

2. **[Nodes]**画面には追加されたノード・リストが表示されます。ノード・リストには、ユーザーがノードを追加しなくてもJEUSをインストールする際に基本的に生成されたノードが照会されます。ノード・リストで、ノード名、タイプ、制御可能有無、ノードにインストールされたJEUSのバージョン情報が確認できます。ノードを追加するために**[Add]**ボタンをクリックします。

参考

ノード設定は、**[LOCK & EDIT]**を使用した設定変更モードへの切り替えや、**[Activate Changes]**を使用した変更の反映のような動作は不要です。

3. **[Nodes]**画面でノード名、ホスト・アドレス、JEUSのインストール・パスを入力します。ノード・タイプに「**Ssh**」を選択して関連項目を入力した後、**[確認]**ボタンをクリックします。
4. **[Nodes]**画面の上部にノードが正常に追加されたとのメッセージと、ノード・リストに追加されたノード情報が表示されます。

リモート・ノードにJEUSのインストールが完了したら、「**JEUS version**」にインストールされたJEUSバージョンが表示されます。JEUSがインストールされていない場合は、「**JEUS version**」にJEUSバージョンが表示されません。制御可能なノードの場合は、**[install]**ボタンをクリックしてJEUSをインストールすることができます。

5. JEUSがインストールされたノードでサーバーを実行するには、上記で定義したノードを当該サーバーにマッピングする必要があります。サーバー設定画面の「**Node Name**」にノード名を指定してサーバーにマッピングします。

参考

同設定は動的変更項目なので、設定を変更する前に**[LOCK & EDIT]**ボタンをクリックして設定変更モードに切り替えます。設定変更が完了したら、**[Activate Changes]**ボタンをクリックして変更内容を反映します。

6. JEUSがインストールされたノードにマッピングされたサーバーは、画面左側の**[Monitoring] > [Servers]**メニューをクリックすると表示されるサーバー・リストにて確認できます。

照会されたサーバー・リストから目的のサーバーを選択した後、**[start]**ボタンをクリックすると表示される画面にて各項目を設定した後、**[確認]**ボタンをクリックすると当該サーバーが開始されます。

参考

サーバーを起動する方法として、**[Monitoring] > [Servers]**メニューの外に、**[Servers]**メニューを使用することもできます。詳しい使用方法については、「[第2章 Javaタイプのノード・マネージャー](#)」を参照してください。

7. リモート・ノードにサーバー実行が完了したら、画面の上部にサーバーが開始されたとのメッセージと、「**Status**」に当該サーバーの状態が「**RUNNING**」で表示されます。

コンソール・ツールの使用

以下は、コンソール・ツールを使用してノードを設定する方法です。

1. **add-ssh-node**コマンドを使って、JEUSにSSHタイプのノードを追加します。

```
[DAS]domain1.adminServer>add-ssh-node node2 -host 192.168.23.129  
-dir /home/jeus/jeus8 -user jeus -privatekey /home/jeus/.ssh/id_rsa  
The node [node2] was successfully added.
```

2. **check-ssh-node**コマンドを使って、追加したSSHノードが正常に動作しているのを確認することができます。当該ノードにSSHを利用してjavaコマンドを実行したとき、以下のようにjavaが実行できると表示されたら、追加したノードが正常に動作していることになります。

```
[DAS]domain1.adminServer>check-ssh-node node2  
The Domain Administration Server can execute the "java" process via SSH.
```

3. 追加されたノードの設定はnodes.xmlに保存されます。**show-node**コマンドを使って設定を確認することができます。

```
[DAS]domain1.adminServer>show-node node2  
=====
```

Property	Value
Node name	node2
Host	192.168.23.129
Mapped servers	
Node Type	SSH
Installed directory	/home/jeus/jeus8
SSH user name	jeus
SSH private key	/home/jeus/.ssh/id_rsa
SSH port	

22

```
+-----+-----+-----+-----+
=====
```

4. このように構成されたノードは、**list-nodes**コマンドを使ってノード・タイプ、制御可能有無、インストールされたJEUSのバージョン情報が確認できます。

```
[DAS]domain1.adminServer>list-nodes
=====
+-----+-----+-----+-----+
| Node name | Type | Under control | JEUS version |
+-----+-----+-----+-----+
| node1     | JAVA | N             | -            |
| node2     | SSH  | Y             | -            |
+-----+-----+-----+-----+
=====
```

上記のようにノードが設定されている場合、コンソール・ツールの**install-jeus**コマンドを使って、当該ノードにJEUSをインストールすることができます。

```
[DAS]domain1.adminServer>install-jeus node2
JEUS was successfully installed on the node [node2].
```

参考

install-jeusコマンドは、DASマシンにインストールされたファイルをコピーする作業なので、当該ノードがDASマシンと同じOSの場合のみ動作できます。OSが異なる場合はnativeライブラリーが使用できないため、正常に動作できません。なお、環境によって実行時間が長くなることもあります。

-
5. サーバーがノードで動作するには、上記のように設定されたノードにサーバーをマッピングする必要があります。**modify-server**コマンドの**-node**オプションを使ってマッピングします。

以下は、サーバーをノードにマッピングする例です。

```
[DAS]domain1.adminServer>modify-server server1 -node node2
Successfully performed the MODIFY operation for server (server1).
Check the results using "list-servers server1 or modify-server server1"
```

6. **show-node**コマンドを使ってノードを確認すると、サーバーがマッピングされていることが確認できます。

```
[DAS]domain1.adminServer>show-node node2
=====
+-----+-----+-----+-----+
| Property | Value |
+-----+-----+-----+-----+
| Node name | node2 |
| Host      | 192.168.23.129 |
+-----+-----+-----+-----+
```


Mapped servers	server1
Node Type	SSH
Installed directory	/home/jeus/jeus8
SSH user name	jeus
SSH private key	/home/jeus/.ssh/id_rsa
SSH port	22
+-----+-----+	
=====	

7. ノードにマッピングされたサーバーは、コンソール・ツールの**start-server**コマンドを使って簡単に開始および終了することができます。なお、サーバーを実行するために必要なすべての情報は自動で設定されます。

```
[DAS]domain1.adminServer>start-server server1
The server [server1] was successfully started.
```

サーバーが動作しているノードでkillコマンドを使って当該サーバーを強制的に終了すると、DASはサーバーがFAILED状態であると判断し、自動的に再起動します。

server-infoコマンドを使って、サーバーの状態変更が確認できます。

```
[DAS]domain1.adminServer>server-info
Information about Domain (domain1)
=====
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Server | Status | Node | PID | Clu | Latest | Need | Listen | Running |
|         |         | Name |     | ster | Start Time | to | Ports | Engines |
|         |         |      |     |      | / Shutdown | Restart |      |          |
|         |         |      |     |      | Time      |      |      |          |
+-----+-----+-----+-----+-----+-----+-----+-----+
| admin | RUNNING | N/A | 104 | N/A | 2016-08-23 | false | base-0.0. | jms, |
| Server | (00:11:1 | 36 |  |  | (火) 午後 |  | 0.0:9736 | ejb, web |
| (*)   | 7)      |  |  |  | 12:44:00 KST |  | http-serv |  |
|         |         |  |  |  |  |  | er-0.0.0.0 |  |
|         |         |  |  |  |  |  | :8088      |  |
|         |         |  |  |  |  |  | jms-0.0.0 |  |
|         |         |  |  |  |  |  | .0:9741    |  |
+-----+-----+-----+-----+-----+-----+-----+-----+
| serve | FAILED( | nod | N/A | N/A | N/A | N/A | N/A | N/A |
| r1    | 00:02:33) | e2  |  |  |  |  |  |  |
+-----+-----+-----+-----+-----+-----+-----+-----+
| serve | SHUTDOWN | N/A | N/A | N/A | N/A | N/A | N/A | N/A |
| r2    |         |  |  |  |  |  |  |  |
+-----+-----+-----+-----+-----+-----+-----+-----+
=====
```

```
[DAS]domain1.adminServer>server-info
Information about Domain (domain1)
=====
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Server | Status | Node | PID | Clu | Latest | Need | Listen | Running |
|         |         | Name |     | ster | Start Time | to | Ports | Engines |
|         |         |      |     |      | / Shutdown | Restart |      |          |
|         |         |      |     |      | Time      |      |      |          |
+-----+-----+-----+-----+-----+-----+-----+-----+
| admin | RUNNING | N/A | 104 | N/A | 2016-08-23 | false | base-0.0. | jms, |
| Server | (00:11:1 | 36 |  |  | (火) 午後 |  | 0.0:9736 | ejb, web |
| (*)   | 7)      |  |  |  | 12:44:00 KST |  | http-serv |  |
|         |         |  |  |  |  |  | er-0.0.0.0 |  |
|         |         |  |  |  |  |  | :8088      |  |
|         |         |  |  |  |  |  | jms-0.0.0 |  |
|         |         |  |  |  |  |  | .0:9741    |  |
+-----+-----+-----+-----+-----+-----+-----+-----+
| serve | FAILED( | nod | N/A | N/A | N/A | N/A | N/A | N/A |
| r1    | 00:02:33) | e2  |  |  |  |  |  |  |
+-----+-----+-----+-----+-----+-----+-----+-----+
| serve | SHUTDOWN | N/A | N/A | N/A | N/A | N/A | N/A | N/A |
| r2    |         |  |  |  |  |  |  |  |
+-----+-----+-----+-----+-----+-----+-----+-----+
=====
```

Server	Status	Node Name	PID	Clu ster	Latest Start Time / Shutdown Time	Need to Restart	Listen Ports	Running Engines
admin Server (*)	RUNNING (00:14:4 8)	N/A	104 36	N/A	2016-08-23 (火) 午後 12:44:00 KST	false	base-0.0. 0.0:9736 http-serv er-0.0.0.0 :8088 jms-0.0.0 .0:9741	jms, ejb, web
serve r1	RUNNING (00:00:1 2)	nod e2	6516	N/A	2016-08-23 (火) 午後 12:58:36 KST	false	BASE-0.0. 0.0:9836	jms, ejb, web
serve r2	SHUTDOWN	N/A	N/A	N/A	N/A	N/A	N/A	N/A

参考

本節で使用されるコマンドの詳しい使用方法については、『*JEUS リファレンスガイド*』の「4.2.3. サーバー管理関連コマンド」, 『*JEUS リファレンスガイド*』の「4.2.4. ドメイン設定関連コマンド」, 『*JEUS リファレンスガイド*』の「4.2.15. ノード管理関連コマンド」を参照してください。

3.3.2. SSHタイプのノード変更

SSHタイプのノードは、WebAdminまたはコンソール・ツールを使用して変更することができます。

1. **modify-ssh-node**コマンドまたは**modify-node**コマンドを使用してノードを変更することができます。

```
[DAS]domain1.adminServer>modify-ssh-node node2 -host 192.168.23.128
The node [node2] was modified successfully. Check the results using "show-node"
[DAS]domain1.adminServer>modify-node node2 -host 192.168.23.127
The node [node2] was modified successfully. Check the results using "show-node"
```

2. 変更されたノード情報を確認するには、**show-node**コマンドを使って該当するノードの設定を照会します。

```
[DAS]domain1.adminServer>show-node node2
=====
+-----+-----+
| Property | Value |
+-----+-----+
```

Node Name	node2	
Host	192.168.23.127	
Mapped Servers	server2	
Node Type	SSH	
Installed directory	/home/jeus/jeus8	
SSH User Name	jeus	
SSH Private Key	/home/jeus/.ssh/id_rsa	
SSH Port		22
+-----+-----+-----+		
=====		

3.3.3. SSHタイプのノード削除

SSHタイプのノードは、WebAdminまたはコンソール・ツールを使用して削除できます。削除方法はJavaタイプのノードと同様なので、本節では省略します。詳細内容については、[「2.4.2. Javaタイプのノード削除」](#)を参照してください。

第4章 RQSプロセス管理

本章では、ノード・マネージャーを利用してRQSを管理する機能について説明します。

4.1. 概要

ノード・マネージャーはサーバー以外のプロセスを管理する機能も持ちます。現在RQSプロセスの管理機能を提供し、今後管理対象が追加される予定です。RQSの管理機能として、RQSプロセスの起動および停止、RQSプロセスの異常終了時のプロセスの再起動をサポートします。このような機能はサーバーやノードの設定とは関係がなく、別途設定された情報により動作します。

参考

RQSプロセスは同じマシン、つまり同じノードにあるノード・マネージャーが管理します。他のマシンにあるRQSプロセスやノード・マネージャーが実行されていないRQSプロセスは管理することができません。

4.2. RQSの動作方式

本節では、ノード・マネージャーを利用したRQSの管理方法について説明します。

4.2.1. RQSプロセスの起動と停止

RQSプロセスは設定ファイルに保存されているファイル・パスとオプションにより起動されます。RQSはサーバーとは独立したプロセスなので、サーバーとは別途に起動および停止します。ノード・マネージャーは、起動時にRQS関連の設定情報を読み込んで、それに従ってRQSプロセスのモニタリングを開始します。このとき、RQSプロセスが起動されていないければ、RQSを起動します。

1. jeusnm.xmlファイルからRQS情報を読み込んで保存します。
2. RQS情報をベースにRQS管理オブジェクトを作成し、モニター用スレッドを作成します。
3. モニタリング・スレッドは読み込んだ情報を利用してプロセスにアクセスした後、モニタリングを開始します。
4. RQSプロセスが異常終了したり、あるいは応答しないことをモニタリングした場合、RQSプロセスを再開します。

参考

プロセスを再起動する際も同様なプロセスで行われます。

4.2.2. RQSプロセスのモニタリング

RQSプロセスのモニタリング時には、別途のファイルを作成しません。プロセスの状態はモニタリング・スレッドとRQSプロセスがやり取りするメッセージに依存します。プロセスにアクセスするときも事前に定義されたメッセージを通じてアクセスし、プロセスの正常動作の判断も周期的にやり取りするメッセージに依存します。またRQSプロセスの正常終了の判断も終了メッセージをやり取りすることで行います。

ノード・マネージャーは周期的にやり取りするメッセージを通じてRQSプロセスの状態をチェックします。この状態チェック・メッセージに対する応答が返されない場合は、何回かメッセージを送り直し、それでも応答が返されない場合は、異常終了したと判断します。

4.2.3. 異常のプロセスの再起動

ノード・マネージャーが周期的に送ったメッセージに数回応答が返されず、RQSプロセスが異常終了したと判断した場合は、プロセスの再起動を行います。プロセスの再起動手順は、まず既存のRQSを確実にキルした後、プロセスを再度開始します。その後、モニタリング・スレッドを再作成して再開始したプロセスのモニタリングを行います。

4.3. 環境設定

ノード・マネージャーがRQSプロセスを管理するには、複数の設定が必要です。RQSプロセスは他のサーバーやマネージャーとは別途に実行されるプロセスであるため、この設定により起動および管理されます。

4.3.1. 設定ファイル

RQSプロセスはサーバーやその他のシステムとは別途に動作するシステムであるため、RQSの起動はjeusnm.xmlファイルの設定に依存します。

RQSプロセスの作成または削除もjeusnm.xmlファイルを直接修正することで行うことができます。jeusnm.xmlのrqsList項目がRQSプロセスの情報を扱っています。

項目	説明
processList	RQSプロセスの情報を含んでいます。 processListの下位のそれぞれのRQS項目が各プロセスの説明を含みます

processList項目の下位には、1プロセスあたり1つのRQS項目が設定されます。複数のプロセスを管理したい場合は、RQS項目を追加することで設定できます。

- RQS

項目	説明
domainName	RQSプロセスのグループ名を設定します。 複数のプロセスはdomainNameという仮想の名前を持つグループに含まれます (必須項目)
processName	RQSプロセスの名前を設定します。 RQSプロセスの開始または停止コマンドを実行するときに使用します(必須項目)
path	RQSプロセスの実行ファイルの場所を設定します。 プロセスが実行されるとき、このパスに設定されたファイルを実行します(必須項目)
port	RQSプロセスのモニタリングに使用されるポート番号を設定します。 RQSプロセスの生存を確認するためのメッセージ交換用ポートです(必須項目)
option	RQSを実行するときに必要な引数を設定します。 RQSマニュアルを参照して必要な機能を設定します
rqmdir	複数のRQSを起動する場合は、それぞれ別々の設定ファイルで実行されます。 それぞれの設定ファイルを認識させる方法として、RQSDIRという環境変数を与える方法がありますが、ここに設定された値を認識して設定ファイルにアクセスできます
retryCount	接続に失敗した場合、何回再接続を試行するか設定します (デフォルト値: 5)
monitoringPeriod	モニタリングに接続を試行するとき、それぞれの接続試行間の時間間隔を設定します(デフォルト値: 500)

参考

上記のとおり、jeusnm.xmlファイルを修正してRQSプロセスの追加または削除が可能です。processListの下位項目としてRQS項目を設けてそれぞれのRQSプロセスの設定値を保存することで、ノード・マネージャがプロセスを管理することができます。

第5章 ノード・マネージャーの冗長化

本章では、ノード・マネージャーが異常終了した場合に、ノード・マネージャーの復旧を可能にする冗長化機能について説明します。

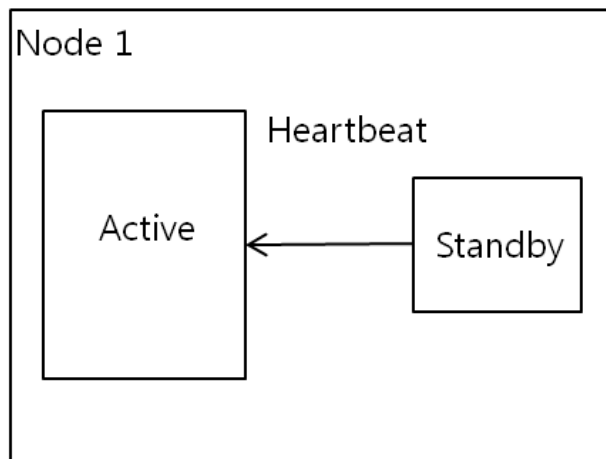
5.1. 概要

ノード・マネージャーはサーバーの起動と終了をサポートし、サーバーに障害が発生した場合は、障害をチェックして再起動する機能を提供します。しかし、ノード・マネージャー自体に異常が発生して終了した場合は、独自で復旧ができません。そのため、ノード・マネージャーの冗長化機能を追加し、ノード・マネージャーが異常終了した場合、あらかじめ実行され待機しているノード・マネージャーが代わりに実行を続けられるようにすることができます。本節では、このノード・マネージャーの冗長化に関する説明と注意点を説明します。

5.2. 冗長化されたノード・マネージャーの動作方式

ノード・マネージャーの冗長化を行うと、アクティブとスタンバイの2つのノード・マネージャーが動作します。このうち、アクティブ・ノード・マネージャーだけが実際にDASやサーバーの要求を処理して状態を管理します。

【図 5.1】 冗長化されたノード・マネージャーの動作方式



スタンバイ・ノード・マネージャーは通常アクティブ・ノード・マネージャーの状態のみをチェックします。そしてアクティブ・ノード・マネージャーの状態チェックで異常を検知すると、アクティブが異常終了したと判断してスタンバイがアクティブの役割を代わって行います。またこのスタンバイ・ノード・マネージャーを監視するためのもう一つのスタンバイ・ノード・マネージャーが実行されます。

1. アクティブ・ノード・マネージャーが開始されると、同時にスタンバイ・ノード・マネージャーが開始されます。
2. スタンバイ・ノード・マネージャーは継続してアクティブ・ノード・マネージャーの状態を監視します。

3. スタンバイ・ノード・マネージャーがアクティブ・ノード・マネージャーの異常を検知します。
4. スタンバイ・ノード・マネージャーはもう一つのスタンバイ・ノード・マネージャーを準備します。
5. スタンバイだった既存のノード・マネージャーがアクティブになってサーバーからの要求を処理します。

ノード・マネージャー状態のチェックに使用するポートは、設定ファイルに設定されたポートを使用します。この値が設定されていない場合は、ノード・マネージャーの冗長化機能を使用しないものと見なします。

参考

スタンバイ・ノード・マネージャーがアクティブ・ノード・マネージャーの異常終了を確認してアクティブになった場合、スタンバイ・ノード・マネージャーを追加で実行して自身を監視するようにします。アクティブ・ノード・マネージャーが異常終了するたびにこの作業は繰り返し行われます。

5.3. 冗長化されたノード・マネージャーの使用方法

ノード・マネージャーの冗長化を使用するには、スタンバイ・ノード・マネージャーとアクティブ・ノード・マネージャーが状態をやり取りするポートを設定する必要があります。このポートは「[2.3.1. 設定ファイル](#)」で説明したように、standbyPort項目を設定することで使用できます。設定されたポートによりアクティブ・ノード・マネージャーとスタンバイ・ノード・マネージャーが通信します。

冗長化を使用したくない場合は、standbyPortを設定しないでください。この値が空の場合、ノード・マネージャーは冗長化機能を使用しないと見なしてスタンバイ・ノード・マネージャーを実行させません。

スタンバイ・ノード・マネージャー

スタンバイ・ノード・マネージャーはアクティブ・ノード・マネージャーの起動時に実行され、アクティブ・ノード・マネージャーの状態を継続してチェックします。

```
[nodemanager-1] [NodeManager-0201] The standby node manager is starting.  
[nodemanager-1] [NodeManager-0102] Initializing the node manager configuration.
```

また、スタンバイ・ノード・マネージャーはサーバーの要求を受けて処理することはないため、別途のログファイルに情報を保存します。このログはノード・マネージャーのログファイルと同じ場所に作成され、ノード・マネージャーの名前の後ろに「_standby」という文字列が付けられたログに保存されます。このログはスタンバイ・ノード・マネージャーだけが使用するログです。

スタンバイ・ノード・マネージャーがアクティブになったときは、スタンバイ用のログにアクティブに切り替わるという内容が記録され、アクティブ用のログを使用するようになります。その後、新しく作成されたスタンバイ・ノード・マネージャーが再びスタンバイ用のログを使用するようになります。そのため、実際のサーバーの要求やサーバー管理ログはアクティブ用のログにだけ残ります。スタンバイ・ログにはスタンバイ・ノード・マネージャーの起動とアクティブ・ノードとの通信ログのみが残ります。

スタンバイ・ノード・マネージャーはアクティブ・ノード・マネージャーと最初に接続するときに、アクティブ・ノード・マネージャーのPIDを受け取ってそれをログに残します。このPIDを確認することで後でどのプロセスがアクティブであるかが分かります。

5.4. 冗長化構成時のノード・マネージャーの終了

ノード・マネージャーを冗長化すると、アクティブ・ノード・マネージャーが異常終了した場合、スタンバイ・ノード・マネージャーがそれをチェックしてノード・マネージャーを継続して起動させます。そのため、ノード・マネージャーを終了するときは、stopNodeManagerスクリプトを利用して終了する必要があります。stopNodeManagerスクリプトは、アクティブ・ノード・マネージャーがスタンバイ・ノード・マネージャーに終了メッセージを送信するようにします。このメッセージを受けたスタンバイ・ノード・マネージャーは接続を整理し、ログに書き込むプロセスまでを行った後、安全に終了します。

参考

問題が発生してノード・マネージャーを強制終了する場合は、スタンバイ・ノード・マネージャーが継続して起動されるため、完璧に終了されないことがあります。そのような場合には、スタンバイ・ノード・マネージャーを先に終了した後、アクティブ・ノード・マネージャーを終了します。ただし、強制終了する場合は、ログが正常に残らないか、他の問題があり得るので、なるべくstopNodeManagerスクリプトを利用するなどの方法を使用します。

索引

シンボル

<serverName>.address, 17
<serverName>.lck, 17
<serverName>.pid, 17
<serverName>.properties, 14, 16
 dasurl, 16
 password, 16
 sslArguments, 16
 username, 16
<serverName>.state, 17

A

actionオプション
 CONTINUE, 13
 ROLLBACK, 13
 STOP, 13
add-java-nodeコマンド, 18
add-ssh-nodeコマンド, 31
apply-patch, 12, 13

C

check-ssh-nodeコマンド, 31
CONTINUE, 13

D

DAS, 2
Domain Administration Server, 2

I

install-jeusコマンド, 32

J

Javaタイプのノード・マネージャー, 2, 9
jeusnm.xml, 14
 host, 15

keystoreFile, 15
keystorePass, 15
logFileName, 16
port, 15
processList, 16, 38
serverAutoRestart, 15
serverMonitoringPeriod, 15
serverRestartDurationTime, 15
serverRestartTryCount, 15
serverRetryRestart, 15
standbyPort, 16
truststoreFile, 15
truststorePass, 15
useNodeManager, 15
useSSLListener, 15

L

list-nodesコマンド, 18, 19, 20, 32

M

modify-java-nodeコマンド, 19, 34
modify-nodeコマンド, 19, 34
modify-serverコマンド, 32

N

nm-connectコマンド, 24
nm-disconnectコマンド, 26
nm-start-serverコマンド, 24
nm-state-serverコマンド, 25
nm-stop-serverコマンド, 25
nm-stopコマンド, 23
Node, 1
Node Manager, 1
nodes.xml, 14, 31
non-interactiveモード, 28

P

processList, 39
 domainName, 39
 monitoringPeriod, 39
 option, 39
 path, 39

port, 39
processName, 39
retryCount, 39
rqmdir, 39

R

remove-nodeコマンド, 20
remove-patch, 12, 13
ROLLBACK, 13

S

server-infoコマンド, 33
show-nodeコマンド, 31, 32
sshコマンド, 28
SSHタイプのノード・マネージャー, 2, 27
start-serverコマンド, 26, 33
startDomainAdminServerNM, 25
startNodeManager, 21
STOP, 13
stopNodeManager, 23

な

ノード, 1
ノード・マネージャー, 1
ノード・マネージャーのRQSプロセス管理, 37
ノード・マネージャーの冗長化によるノード・マネージャー
の復旧, 41

ら

ローリング・パッチ, 12