

JP1 Version 11

ジョブ管理 基本ガイド (スクリプト言語編)

概説・手引書

3021-3-B31

前書き

■ 対象製品

P-8112-B1BL JP1/Advanced Shell 11-00 (適用 OS : Red Hat Enterprise Linux Server 6 (64-bit x86_64), Red Hat Enterprise Linux Server 7 (64-bit x86_64), Oracle Linux 6 (x64), Oracle Linux 7, CentOS 6, CentOS 7, SUSE Linux 12)

P-2A12-B1BL JP1/Advanced Shell 11-00 (適用 OS : Windows 10, Windows Server 2012, Windows 8, Windows 7, Windows Server 2008)

本マニュアルは、JP1 製品のサブカテゴリ「スクリプト言語」のうち、JP1/Advanced Shell について説明します。JP1/Script については説明しません。

■ 輸出時の注意

本製品を輸出される場合には、外国為替及び外国貿易法の規制並びに米国輸出管理規則など外国の輸出関連法規をご確認の上、必要な手続きをお取りください。なお、不明な場合は、弊社担当営業にお問い合わせください。

■ 商標類

HITACHI, Job Management Partner 1, JP1 は、株式会社 日立製作所の商標または登録商標です。

Linux は、Linus Torvalds 氏の日本およびその他の国における登録商標または商標です。

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

Oracle と Java は、Oracle Corporation 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。

Red Hat は、米国およびその他の国で Red Hat, Inc. の登録商標もしくは商標です。

SUSE は、米国およびその他の国における SUSE LLC の登録商標または商標です。

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。

Windows は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

Windows 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または商標です。

その他記載の会社名、製品名などは、それぞれの会社の商標もしくは登録商標です。

■ 発行

2016 年 1 月 3021-3-B31

■ 著作権

All Rights Reserved. Copyright (C) 2016, Hitachi, Ltd.

はじめにお読みください

■ JP1/Advanced Shell でできること

ここでは、JP1/Advanced Shell でできることを説明します。

JP1/Advanced Shell の目的

JP1/Advanced Shell は、バッチ業務の開発生産性や運用効率を向上するための製品です。バッチジョブのためのジョブ定義スクリプト（シェルスクリプト）を効率的に作成・実行できます。

JP1/Advanced Shell には、次に示す特長があります。

バッチ業務の OS 間での資産の継承

- 既存資産の活用

UNIX 環境で作成したシェルスクリプトを利用して Windows 環境でジョブ定義スクリプトを開発できます。

JP1/Advanced Shell で使用するジョブ定義スクリプトでは、シェル標準互換の言語仕様を採用しています。したがって、習得しやすく、既存のシェルスクリプトからの移行も容易です。

- クロスプラットフォームの対応

クロスプラットフォームとは、複数の OS 基盤のことです。クロスプラットフォームに対応した機能が利用できます。

- Windows 環境で開発したジョブ定義スクリプトを Windows 環境でも UNIX 環境でも実行できます。
- UNIX 互換コマンドを、Windows 環境でも UNIX 環境でも使用できます。

バッチ業務の構築のスピードアップ

- ジョブの実行の制御

JP1/Advanced Shell では、バッチ業務で繰り返し使用される処理を自動化したり、簡潔に記述したりできるようにジョブ定義スクリプトを拡張しています。

次の機能を使用してジョブ定義スクリプトの記述量を削減し、ジョブ定義スクリプトの可読性や保守性を向上できます。

- ジョブステップの実行条件を指定できます。
- ジョブステップ内で有効な変数を使用できます。
- バッチジョブがエラー終了した場合、エラーメッセージを出力したり、終了コードを設定したりできます。

- バッチジョブがエラー終了した場合、自動的に子プロセスを強制終了して、バッチジョブで使用した一時ファイルを自動的に削除できます。

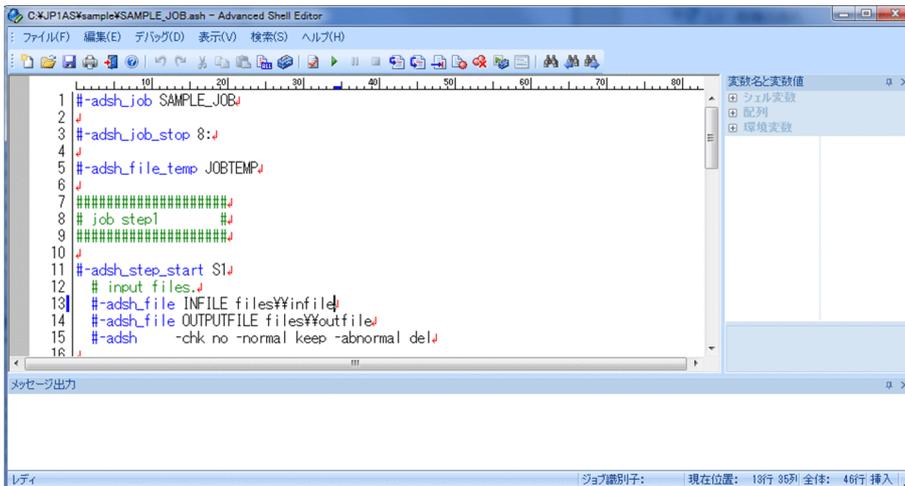
• エディタでのジョブ定義スクリプトの開発（開発環境）

GUI（Graphical User Interface）の JP1/Advanced Shell エディタ（デバッグ機能付き専用エディタ）を使用した開発環境でジョブ定義スクリプトを開発し、デバッグできます。

- ジョブステップ単位で実行したり、ブレークポイントを設定したりできます。
- ジョブ定義スクリプトのカバレッジ情報を蓄積できます。

JP1/Advanced Shell エディタウィンドウを次の図に示します。

JP1/Advanced Shell エディタウィンドウ



• ファイルの割り当ておよび後処理の効率化

通常ファイルの存在チェックや一時ファイルの割り当てと削除などの処理を自動化でき、簡潔に記述できます。

- バッチジョブの実行中に自動的に一時ファイルを割り当て、バッチジョブ終了時に削除できます。
- バッチジョブの実行中に通常ファイルの存在チェック、およびジョブステップまたはジョブの結果によるファイルの後処理ができます。

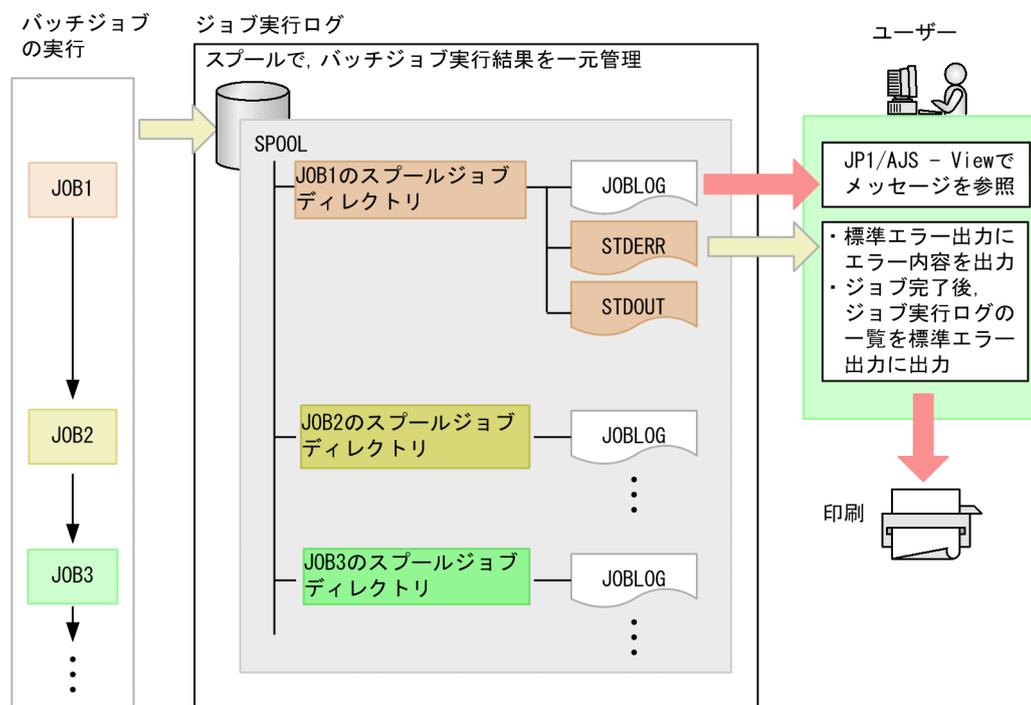
バッチジョブの実行結果の一元管理による運用性・保守性の向上

ジョブ実行ログを自動的に出力し、一元管理することで、障害時などの保守性を向上できます。

従来、オープンシステムでのバッチジョブの実行結果は、格納先が一元化されていなかったため管理が煩雑でした。JP1/Advanced Shell で運用した場合、ジョブ実行ログを採取することで、バッチジョブの実行結果をプールに集めて一元管理できます。また、JP1/AJS - View を使用することで、ジョブ定義スクリプトの実行を自動化して定期的にバッチジョブを実行したり、バッチジョブの実行結果も参照したりできます。

各ジョブの実行結果は、スプールディレクトリの下にスプールジョブのディレクトリとして出力されます。バッチジョブの実行結果の一元管理を次の図に示します。

バッチジョブの実行結果の一元管理



ジョブ実行ログの出力内容については、「4.2.1 ジョブ実行ログ」を参照してください。

業務への応用例

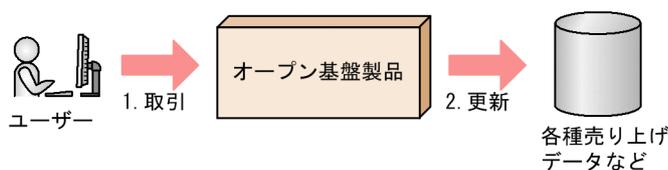
JP1/Advanced Shell は、次のような業務に応用できます。

日中にオンラインシステムで多数の取引があり、夜間にその集計をする業態であれば、例えば、売り上げや商品の販売数、在庫数などを集計するバッチジョブを開発し、実行できます。また、日次処理、月次処理、期末処理といった定型集計用だけでなく、特定の用途や、臨時のタイミングで使用するバッチジョブも開発・実行できます。

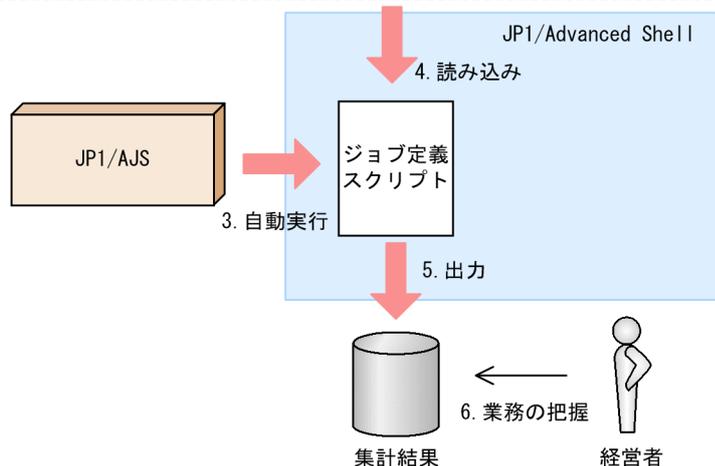
JP1/Advanced Shell の運用例（日々の営業集計の場合）を次の図に示します。

JP1/Advanced Shell の運用例（日々の営業集計の場合）

●日々の営業の開始



●日々の営業の終了



1. 日々の営業を開始し、ユーザーは商品などの取引をします。
2. オープン基盤製品は、各種売り上げデータなどを更新します。
3. 日々の営業が終了して、JP1/AJS は指定された時間にジョブ定義スクリプトの自動実行を指示します。
4. JP1/Advanced Shell は各種売り上げデータを処理するためのジョブ定義スクリプトを実行します。
5. JP1/Advanced Shell はジョブ定義スクリプトの実行結果を出力します。
6. 経営者は、実行結果を基に集計情報や商品の売り上げの推移などを把握できます。

■ このマニュアルで説明すること

このマニュアルは、JP1/Advanced Shell の基本的な構築方法および運用方法について説明しています。このマニュアルを読んだ方が、JP1/Advanced Shell の概念や基本的な使い方をひとつお理解できるようになることを目的としています。

このマニュアルは次の方にお読みいただくことを前提に説明します。

- JP1/Advanced Shell の導入を検討している方
- JP1/Advanced Shell の概念、構築から運用までの操作感を短時間で把握したい方

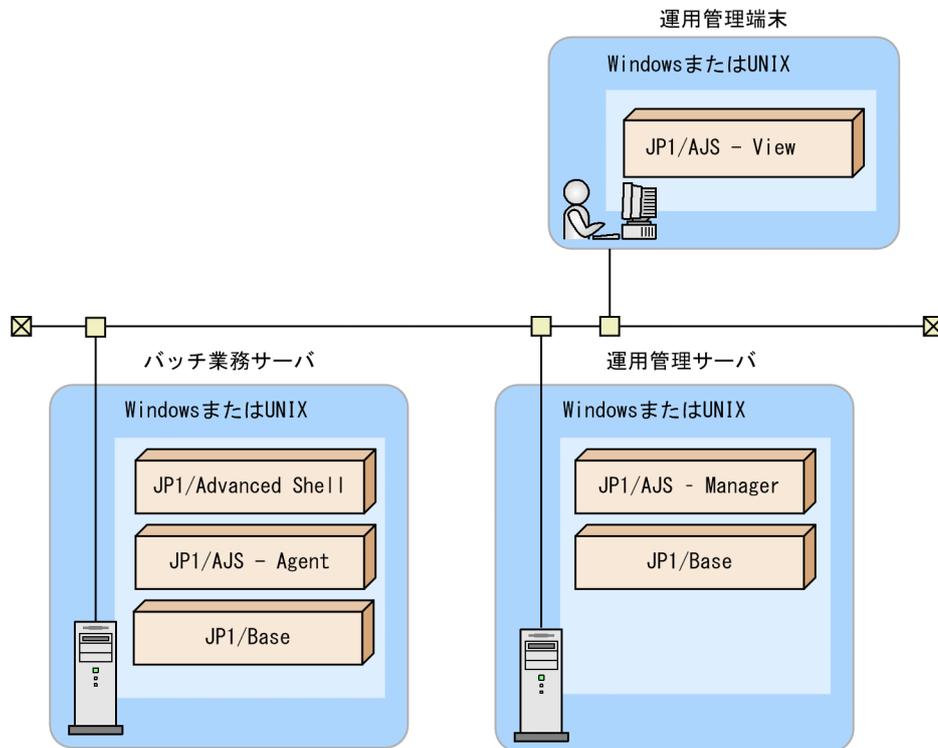
JP1/Advanced Shell は次の製品で構成されます。

- JP1/Advanced Shell（バッチジョブ向けスクリプト実行基盤）

- JP1/Advanced Shell - Developer (バッチジョブ向けスクリプト開発基盤)

基本ガイドでは、これらの製品のうち、JP1/Advanced Shell を対象に説明します。

このマニュアルでは、次のシステム構成を前提に説明します。ここで示す構成以外での運用については、マニュアル「JP1/Advanced Shell」のシステム構成の説明を参照してください。



システムの各構成要素の役割を次に説明します。

- バッチ業務サーバ：ジョブ定義スクリプトを手動または自動実行します。
- 運用管理サーバ：実行されたジョブを管理します。
- 運用管理端末：JP1/AJS - View を使用してジョブの実行結果を表示したり、自動実行するジョブ定義スクリプトを定義したりします。

■ マニュアルの読み方

JP1/Advanced Shell のマニュアルには、この基本ガイドのほかに、マニュアル「JP1/Advanced Shell」があります。基本ガイドで説明していないシステム構成の構築手順や運用手順の詳細、コマンドやメッセージといったリファレンス情報などについては、そちらをお読みください。

なお、このマニュアルでは Windows と Linux の手順を記載します。本文中に UNIX 環境の説明がありますが、それらは「UNIX」を「Linux」に読み替えてください。ただし、「UNIX 互換コマンド」と「UNIX ジョブ」については、機能名のため、「Linux」への読み替えは不要です。

他マニュアルへの参照先は、『～については、マニュアル「△△」の「○○」の説明を参照してください。』の形式で記載しています。「○○」をキーワードとしてマニュアル「△△」内を検索して、該当する説明をお読みください。

製品の改良などによって、このマニュアルに掲載されている画面はご使用の製品画面と一部異なることがあります。あらかじめご了承ください。

目次

前書き 2

はじめにお読みください 3

1 JP1/Advanced Shell および前提製品のインストール 11

1.1 JP1/Advanced Shell および前提製品のインストールの流れ 12

1.1.1 インストールの流れ (Windows の場合) 12

1.1.2 インストールの流れ (Linux の場合) 12

1.2 インストール前の準備 13

1.2.1 前提 OS 13

1.2.2 インストールに必要なメモリ, ディスク容量 13

1.2.3 前提となる OS の言語設定 13

1.2.4 JP1/Advanced Shell で使用できるファイルを確認する 14

1.2.5 ローカルタイムの設定を確認する 16

1.3 前提製品のインストール 17

1.3.1 JP1/AJS とその関連製品をインストールする 17

1.4 JP1/Advanced Shell のインストール 18

1.4.1 JP1/Advanced Shell をインストールする (Windows の場合) 18

1.4.2 JP1/Advanced Shell をインストールする (Linux の場合) 18

2 JP1/Advanced Shell の環境構築 21

2.1 環境ファイルを設定する 22

2.1.1 システム環境ファイルを設定する 22

2.1.2 ジョブ環境ファイルを設定する 22

2.1.3 必要なディレクトリを作成する 23

2.1.4 UNIX 互換コマンドを使用するための設定をする 25

2.1.5 JP1/AJS からジョブを起動するためのシェルを設定する (Linux の場合) 26

2.1.6 子孫ジョブとして起動するファイルを定義する 27

2.1.7 ジョブの種類ごとのジョブ実行ログの出力内容 28

2.1.8 ジョブ実行結果とログの出力情報を定義する 30

2.1.9 スクリプト拡張コマンドの終了コードを定義する 35

2.1.10 ジョブ強制終了時にユーザー固有の後処理を実行する 35

2.2 環境変数を設定する 37

2.2.1 ジョブを続行できないエラーが発生したときの終了コードを定義する 37

3	JP1/AJS のジョブ定義 39
3.1	JP1/AJS でジョブを定義する 40
3.1.1	ジョブネットを定義して実行する 40
3.1.2	PC ジョブでバッチジョブを定義する 43
3.1.3	UNIX ジョブでバッチジョブを定義する 44
4	バッチジョブの実行と実行結果の確認 48
4.1	バッチジョブの実行 49
4.1.1	バッチジョブの実行の流れ 49
4.1.2	バッチジョブを実行する 49
4.2	バッチジョブの実行結果の確認 53
4.2.1	ジョブ実行ログ 53
4.2.2	バッチジョブの実行結果を確認する 54
4.2.3	スプールジョブを削除する 57
4.3	バッチジョブの強制終了 59
4.3.1	子プロセスまたは子孫プロセスの強制終了 60
4.3.2	Ctrl+C などの操作に関する注意事項 (Linux の場合) 60

付録 61

付録 A	もっと使いこなすには? 62
付録 B	このマニュアルの参考情報 63
付録 B.1	関連マニュアル 63
付録 B.2	マイクロソフト製品の表記 63
付録 B.3	このマニュアルで使用している記号 64
付録 B.4	Windows の場合の「インストール先フォルダ」について 65
付録 B.5	このマニュアルで使用する「共通アプリケーションフォルダ」について 65
付録 B.6	このマニュアルで使用する「共有ドキュメントフォルダ」について 66
付録 B.7	このマニュアルで使用する Windows のメニュー名について 66
付録 B.8	フォルダおよびディレクトリの表記 66
付録 B.9	製品名の表記 66
付録 B.10	KB (キロバイト) などの単位表記 67
付録 C	用語解説 68

索引 75

1

JP1/Advanced Shell および前提製品のインストール

ここでは、JP1/Advanced Shell とその前提製品のインストール方法について説明します。

1.1 JP1/Advanced Shell および前提製品のインストールの流れ

JP1/Advanced Shell および前提製品のインストールの流れについて説明します。

1.1.1 インストールの流れ (Windows の場合)

ここでは、Windows 環境で、JP1/Advanced Shell と関連する前提製品をインストールする流れについて説明します。

製品および関連プログラムのインストールの流れを次に示します。

1. 運用管理サーバで必要な製品のインストールとセットアップを実施する。
2. 運用管理端末で必要な製品のインストールとセットアップを実施する。
3. バッチ業務サーバで必要な製品のインストールとセットアップを実施する。
4. バッチ業務サーバで JP1/Advanced Shell のインストールと環境情報の設定などをする。

1.1.2 インストールの流れ (Linux の場合)

ここでは、Linux 環境で、JP1/Advanced Shell と関連する前提製品をインストールする流れについて説明します。

製品および関連プログラムのインストールの流れを次に示します。

1. 運用管理サーバで必要な製品のインストールとセットアップを実施する。
2. Windows 環境の運用管理端末で必要な製品のインストールとセットアップを実施する。
3. バッチ業務サーバで必要な製品のインストールとセットアップを実施する。
4. バッチ業務サーバに JP1/Advanced Shell のインストールと環境情報の設定などをする。

1.2 インストール前の準備

ここでは、JP1/Advanced Shell と前提製品をインストールする前に必要な準備について説明します。

1.2.1 前提 OS

- Red Hat Enterprise Linux Server 6 (64-bit x86_64)
- Red Hat Enterprise Linux Server 7 (64-bit x86_64)
- Oracle Linux 6 (x64)
- Oracle Linux 7
- CentOS 6 (x64)
- CentOS 7
- SUSE Linux 12
- Windows Server 2008
- Windows 7
- Windows 8
- Windows Server 2012
- Windows 10

1.2.2 インストールに必要なメモリ，ディスク容量

JP1/Advanced Shell のインストールに必要なメモリ，ディスク容量については，計算式を使って値を算出する必要があります。詳細については，マニュアル「JP1/Advanced Shell」の「メモリ所要量およびディスク占有量」の説明を参照してください。

1.2.3 前提となる OS の言語設定

JP1/Advanced Shell でサポートされている言語設定およびエンコーディングを，OS ごとに示します。JP1/Advanced Shell で使用するファイルの言語およびエンコーディングは，JP1/Advanced Shell が動作する環境の言語設定およびエンコーディングと一致させてください。一致していない場合，文字化けが発生したり，入出力データの内容が不当になってしまったりすることがあるため，必ず確認してください。

表 1-1 OS ごとの言語設定およびエンコーディング

OS	言語設定	エンコーディング
Windows	表示言語が「日本語」、システムロケールの設定が「日本語(日本)」になっている。	Shift-JIS
Linux	LANG 環境変数の値が「ja_JP.UTF-8」または「C」になっている。	UTF-8 C

1.2.4 JP1/Advanced Shell で使用できるファイルを確認する

ここでは、JP1/Advanced Shell で使用するファイルの一覧、ファイルやパスの指定に関する注意事項について説明します。

(1) JP1/Advanced Shell で使用するファイルの一覧

JP1/Advanced Shell で使用するファイルを次の表に示します。ファイルサイズが 2GB を超えられるかどうかは、マニュアル「JP1/Advanced Shell」の「JP1/Advanced Shell で使用するファイル」の説明を参照してください。

表 1-2 JP1/Advanced Shell で使用するファイル

ファイル名 (アイコン)	拡張子	ファイルの内容
ジョブ定義スクリプト ファイル ()	.ash	ジョブ定義スクリプトを保存するファイルです。ファイル名は、ユーザー任意に指定できます。
環境ファイル*	.ase	JP1/Advanced Shell の環境情報を設定するファイルです。
システム環境ファイル	.ase	システムの環境情報を設定するファイルです。
カバレッジ情報ファイル	.asc	JP1/Advanced Shell のカバレッジ環境情報です。
デバッグ情報ファイル	.asd	エディタ（開発環境）で使用するデバッグ情報です。
システム実行ログ*	.log	バッチジョブの実行履歴を統括的に参照するためのシステム管理者向けのログ情報です。
トレースログ*	.log	JP1/Advanced Shell の内部トレースログです。
一時ファイル	.tmp	システム内部で使用する一時ファイルです。
カバレッジ表示一時ファイル	.txt	カバレッジ情報の表示で使用する一時ファイルです。ファイル名の形式を次に示します。 <div style="border: 1px solid black; padding: 2px; margin-top: 5px;"> adshexec_view_ジョブ定義スクリプトファイル名_年月日_時分秒.txt </div>
起動ログ【UNIX 限定】	.log	ユーザー応答機能管理デーモンの起動・停止時に採取されるログ情報です。

ファイル名 (アイコン)	拡張子	ファイルの内容
pid ファイル【UNIX 限定】	.pid	ユーザー応答機能管理デーモンと adshmdctl コマンドで使用するファイルです。
アプリケーション実行エージェント機能ログ※ 【Windows 実行環境限定】	.log	アプリケーション実行エージェント機能の内部ログです。

注※

これらのファイルは adshcollect コマンドで採取できます。採取方法については、マニュアル「JP1/Advanced Shell」の「adshcollect コマンド」の説明を参照してください。

ファイルとパスの指定に関する注意事項

- ディレクトリ区切り文字は、Windows であれば「¥」※、UNIX であれば「/」を使用できます。それ以外の文字を使用した場合の動作は保証できません。

UNIX のディレクトリ区切り文字に「¥」を使用した場合、ディレクトリ区切り文字とはみなされず、正しく動作しません。

Windows のディレクトリ区切り文字に「/」を使用した場合、ディレクトリ区切り文字とみなされることがあります。ただし、使用方法によってはディレクトリ区切り文字とみなされず、正しく動作しないこともあります。

注※

ジョブ定義スクリプトに記述した「¥」はエスケープ文字と見なされるため、「¥¥」と記述するか、シングルクォーテーションで囲んでください。

- ファイル名に.（ドット）で始まる名称を使用しないでください。
- 指定できるパス名の最大長は、使用している OS の仕様に従います。
- ファイル名は最大 246 バイトです。【Windows 限定】
- ファイル名に予約デバイス名（CON, AUX, NUL など）は使用しないでください。【Windows 限定】
- ファイル名に NTFS のストリームは使用しないでください。【Windows 限定】
- ジャンクションは使用しないでください。【Windows 限定】
- ファイル名やパス名には UNC 形式の名称（例：¥コンピュータ名¥共有名¥ファイル名）を使用できますが、パス名の最後が「共有名」（または「共有名¥」）にならないように指定してください。また、シェル標準コマンドの cd コマンドには UNC 形式を指定できません。【Windows 限定】

【使用できる UNC 形式】

¥server¥share¥dir

¥10.111.222.33¥share¥dir

【使用できない UNC 形式】

¥server¥share

¥10.111.222.33¥share

- トレース、システム実行ログ、スプールおよび一時ファイルのフォルダパス名に UNC 形式の名称は使用しないでください。【Windows 限定】

(2) ファイルシステムに関する注意事項

JP1/Advanced Shell で使用する場合に注意が必要なファイルシステムを次に示します。

- NFS
サポートしません。
- HSFS
HSFS を使用する場合、次の注意事項があります。
 - HSFS 上に JP1/Advanced Shell をインストールできません。
 - HSFS 上にシステム実行ログおよびトレースを作成できません。
 - ユーザー応答機能を使用する場合、スプールジョブディレクトリを HSFS 上に設定できません。
 - HSFS 07-00 より前のバージョンを使用する場合、UNIX 互換コマンドを使って HSFS 上のファイルおよびディレクトリを参照・更新するには、事前に HSFS のシステムオプションである CPFS_CACHE_POLICY に NOCACHE を指定する必要があります。
 - HSFS 07-00 以降のバージョンを使用する場合、UNIX 互換コマンドを使って HSFS 上のファイルおよびディレクトリを参照・更新するには、事前に HSFS のシステムオプションである CPFS_COMPAT_LINKCNT=0 を指定する必要があります。デフォルトは指定されている状態です。

1.2.5 ローカルタイムの設定を確認する

JP1/Advanced Shell では、ローカルタイム情報を環境変数から参照して情報を入出力するため、それらを環境変数で事前に設定しておく必要があります。

JP1/Advanced Shell が提供するコマンドは、OS のタイムゾーンの設定 (Windows の場合)、または環境変数 TZ (UNIX の場合) に従って情報を出力します。環境変数 TZ は、次に示すどれかの方法で指定してください。

- JP1/AJS のジョブ定義、環境変数の定義で指定
- システムプロファイル (/etc/profile) に指定
- ユーザープロファイル (\$HOME/.profile) に指定

1.3 前提製品のインストール

ここでは、前提製品である JP1/AJS と、その関連製品のインストールについて説明します。

1.3.1 JP1/AJS とその関連製品をインストールする

JP1/AJS とその関連製品を次の表に示します。

表 1-3 JP1/AJS とその関連製品

サーバの種類	実施する処理	プログラム
JP1/Advanced Shell と同じバッチ業務サーバ	JP1/AJS からジョブ定義スクリプトを実行する	JP1/Base JP1/AJS - Agent [※]
運用管理サーバ	ジョブを管理する	JP1/Base JP1/AJS - Manager [※]
運用管理端末【Windows 限定】	ジョブの実行結果を表示する	JP1/AJS - View

注※

JP1/AJS - Manager には JP1/AJS - Agent の機能が含まれているため、JP1/Advanced Shell と同一サーバ上に JP1/AJS - Manager があれば、JP1/AJS - Agent は不要です。

JP1/AJS と JP1/Base のインストール方法の詳細については、マニュアル「ジョブ管理 基本ガイド（ジョブスケジューラー編）」を参照してください。

1.4 JP1/Advanced Shell のインストール

ここでは、JP1/Advanced Shell をインストールする手順について説明します。インストールは、管理者権限を持つユーザーで実行してください。

1.4.1 JP1/Advanced Shell をインストールする (Windows の場合)

JP1/Advanced Shell を新規インストールする手順を次に示します。

1. JP1/Advanced Shell をインストールする Windows マシンに管理者権限を持つユーザーでログオンする。
2. すべてのプログラムを終了する。
3. JP1/Advanced Shell の CD-ROM 媒体を CD-ROM ドライブに入れる。
4. 起動したインストーラの指示に従って必要な情報を入力し、インストールする。
インストール時に定義する情報を次に示します。
 - インストールする製品 (JP1/Advanced Shell) の選択
 - ユーザー情報
 - インストール先フォルダ
5. [完了] ダイアログが表示されたら、[完了] をクリックする。
インストールが完了します。

1.4.2 JP1/Advanced Shell をインストールする (Linux の場合)

CD-ROM 媒体を使った Linux へのインストール手順を次に示します。

なお、CD-ROM のディレクトリ名やファイル名は、ハードウェア環境などによって記述した内容と見え方が異なります。ls コマンドで確認し、表示されたファイル名をそのまま入力してください。

1. ユーザー権限を設定する。
JP1/Advanced Shell をインストールするサーバに、スーパーユーザーでログインします。または、su コマンドでユーザー権限をスーパーユーザーに変更します。
2. すべてのプログラムを終了する。
既存の JP1 シリーズのプログラム、および JP1/Advanced Shell のプログラムが動作している場合、必ず停止させます。
3. JP1/Advanced Shell の媒体をセットする。
4. 次のコマンドを実行して、CD-ROM 装置をマウントする。

```
/bin/mount -r -o mode=0544 /dev/cdrom /cdrom
```

/cdrom は CD-ROM デバイススペシャルファイルのマウントポイントです。マウントポイントディレクトリがない場合は、作成してください。なお、デバイススペシャルファイル名およびマウントポイントは、使用する環境によって異なる場合があります。

5. 次のコマンドを実行して、Hitachi PP Installer を起動する。

```
/cdrom/LINUX/setup /cdrom*
```

注※ ここでは、マウントポイントに/cdrom を仮定します。

Hitachi PP Installer が起動し、初期画面が表示されます。

Hitachi PP Installer の初期画面の例を次に示します。

```
Hitachi PP Installer 05-24
```

```
L) List Installed Software.  
I) Install Software.  
D) Delete Software.  
Q) Quit.
```

```
Select Procedure ==>
```

```
+-----+  
CAUTION!  
YOU SHALL INSTALL AND USE THE SOFTWARE PRODUCT LISTED IN THE  
"List Installed Software." UNDER THE TERMS AND CONDITION OF  
THE SOFTWARE LICENSE AGREEMENT ATTACHED TO SUCH SOFTWARE PRODUCT.  
+-----+
```

```
All Rights Reserved. Copyright (C) 1994, 2015, Hitachi, Ltd.
```

6. Hitachi PP Installer の初期画面で「I」を入力する。

インストールできるプログラムの一覧が表示されます。

7. JP1/Advanced Shell を選択して「I」を入力する。

JP1/Advanced Shell がインストールされます。なお、プログラムを選択するには、カーソルを移動させてスペースバーで選択します。

PP インストール画面の例を次に示します。

```
      PP-No.          VR      PP-NAME
<@>001 P-8112-B1BL    1100    Advanced Shell
:
:
F) Forward B) Backward J) Down K) Up Space) Select/Unselect I) Install Q) Quit
```

選択した PP の左側に、<@>が表示されます。続いて [I] を入力すると、最下行に次に示すメッセージが表示されます。

```
Install PP? (y: install, n: cancel) ==>
```

ここで、[y] または [Y] を選択するとインストールが開始されます。[n] または [N] を選択すると、インストールが中止され、PP インストール画面に戻ります。

8. インストールが正常終了したら、「Q」を入力する。

Hitachi PP Installer の初期画面に戻ります。

なお、インストール時にインストーラのログとして次のファイルが作成されます。

```
/opt/jp1as/instlog/ADSH_INST_LOG
/opt/jp1as/instlog/ADSH_INST_USERLOG
```

インストーラのログファイルが作成されていない場合、次に示す問題が考えられます。

- インストーラのログファイルが通常のファイルでない
- インストーラのログファイルを作成するディレクトリに書き込み権限がない
- インストーラのログファイルのパス名を構成する各ファイルパスに、同一のファイルが存在する同一のファイルが存在する状態を次に示します。
 - "/opt"がディレクトリでない
 - "/opt/jp1as"がディレクトリでない
 - "/opt/jp1as/instlog"がディレクトリでない

2

JP1/Advanced Shell の環境構築

ここでは、JP1/Advanced Shell の環境を構築する手順について説明します。

2.1 環境ファイルを設定する

環境ファイルには、システム環境ファイルとジョブ環境ファイルがあります。設定できるパラメーターは同じです。各ファイルの説明を次の表に示します。

表 2-1 環境ファイルの種類

環境ファイルの種類	説明
システム環境ファイル	システム管理者が設定する、システム共通の環境ファイルです。固定のディレクトリに格納することで自動的に使用される環境ファイルです。
ジョブ環境ファイル	開発者がジョブごとに設定する環境ファイルです。次の場合に指定します。 <ul style="list-style-type: none">環境変数 ADSSH_ENV で指定する環境ファイル

ジョブコントローラは、システム環境ファイルおよびジョブ環境ファイルの設定内容を使用します。

システム環境ファイルのパラメーターに指定する各種ディレクトリは、存在するディレクトリを指定する必要があります。そのため、デフォルトのディレクトリを変更する場合は、変更後のディレクトリをあらかじめ作成してください。

また、UNIX 環境でシステム環境ファイルを変更したあとは、必ず `adshmdctl` コマンドに `conftest` オプションを指定して実行し、エラーのないことを確認してください。

それぞれの環境ファイルの設定方法を次に示します。

2.1.1 システム環境ファイルを設定する

システム環境ファイルはシステム管理者が作成および設定します。作成したシステム環境ファイルは、次の表に示すファイルパスに格納することで有効になります。

表 2-2 システム環境ファイルのファイル名

環境	システム環境ファイルのファイル名
Windows (実行環境)	共通アプリケーションフォルダ\HITACHI\JP1AS\JP1ASE\conf\adshrc.ase
UNIX	/opt/jplas/conf/adshrc.ase

2.1.2 ジョブ環境ファイルを設定する

ジョブ環境ファイルを使用してバッチジョブを実行する場合は、環境変数 `ADSSH_ENV` にファイルパスを設定します。次に示す手順でジョブ環境ファイルを作成して設定してください。

1. 次のディレクトリにある環境ファイルのサンプルデータ `sample.ase` を、任意のディレクトリ・ファイル※にコピーする。

- Windows の実行環境の場合
インストール先フォルダ¥JP1ASE¥sample¥sample.ase
- UNIX の実行環境の場合
/opt/jp1as/sample/sample.ase

2. コピーしたジョブ環境ファイルに必要なパラメーターを定義する。

3. 作成したジョブ環境ファイルをバッチジョブ実行時に使用するよう、作成したジョブ環境ファイルのパスを環境変数 ADSSH_ENV に設定する。

環境変数 ADSSH_ENV は、次に示すどれかの方法で指定します。

- 【Windows 限定】 OS の設定で環境変数を指定する
- 【UNIX 限定】 システムプロファイル/etc/profile を指定する
- 【UNIX 限定】 ユーザープロファイル (\$HOME/.profile) を指定する

注※

ジョブ環境ファイルの任意のディレクトリ名およびファイル名に&, (,), [,], {, }, ^, =, ;, !, ', +, ,, ` , ~, #, %の記号を含めないでください。これらの記号を含む場合、正常に動作しません。

2.1.3 必要なディレクトリを作成する

JP1/Advanced Shell のインストールが完了したあと、実行に必要なディレクトリをデフォルト設定から変更する場合、変更後のディレクトリを作成し、環境ファイルに指定します。

JP1/Advanced Shell で必要なディレクトリの種類と指定内容を次に示します。JP1/Advanced Shell を実行するユーザーは、これらのディレクトリに対して必要な権限を割り当ててください。

- 一時ファイル用のディレクトリ
バッチジョブ内だけで利用するファイルを一時的に作成するディレクトリを指定します。
- スプール用のディレクトリ
ジョブ実行ログやプログラム出力データファイルを格納するディレクトリを指定します。
- システム実行ログ用のディレクトリ
システム管理者がバッチジョブの実行を監視するために、バッチジョブの履歴をシステム実行ログとして保存するディレクトリを指定します。
- トレース用のディレクトリ
システム障害時に、トラブルシューティングとして状況を保存するためのディレクトリを指定します。

JP1/Advanced Shell で必要なディレクトリを次に示します。

表 2-3 JP1/Advanced Shell で必要なディレクトリ

ディレクトリの種類	環境設定パラメーター	デフォルトディレクトリまたはパス	デフォルトの権限
一時ファイル用のディレクトリ	TEMP_FILE_DIR	<ul style="list-style-type: none"> 実行環境【Windows 限定】 共有ドキュメントフォルダ¥Hitachi¥JP1AS¥JP1ASE¥temp 実行環境【UNIX 限定】 /var/opt/jp1as/temp 	CRWD【Windows 限定】 1777【UNIX 限定】
スプール用のディレクトリ	SPOOL_DIR	<ul style="list-style-type: none"> 実行環境【Windows 限定】 共有ドキュメントフォルダ¥Hitachi¥JP1AS¥JP1ASE¥spool 実行環境【UNIX 限定】 /var/opt/jp1as/spool 	CRWD【Windows 限定】 1777【UNIX 限定】
システム実行ログ用のディレクトリ	LOG_DIR LOG_FILE_CNT LOG_FILE_SIZE	<ul style="list-style-type: none"> 実行環境【Windows 限定】 共有ドキュメントフォルダ¥Hitachi¥JP1AS¥JP1ASE¥log 実行環境【UNIX 限定】 /opt/jp1as/log 	CRWD【Windows 限定】 0777【UNIX 限定】
トレース用のディレクトリ	TRACE_DIR TRACE_FILE_CNT TRACE_FILE_SIZE TRACE_LEVEL	<ul style="list-style-type: none"> 実行環境【Windows 限定】 共通アプリケーションフォルダ¥Hitachi¥JP1AS¥JP1ASE¥trace 実行環境【UNIX 限定】 /opt/jp1as/trace 	CRWD【Windows 限定】 1777【UNIX 限定】

(凡例)

次に示すデフォルトの権限の英字は、権限の種類を示します。

C：作成，R：読み込み，W：書き込み，D：削除

(1) 必要な権限

バッチジョブを実行するユーザーに対して、必要な権限を次に示します。

(a) Windows の場合

バッチジョブを実行するユーザーに対して、フルコントロールを設定してください。

(b) UNIX の場合

バッチジョブを実行するユーザーに対して、各ディレクトリには次に示すファイルパーミッションが必要です。

表 2-4 ディレクトリのファイルパーミッション

ディレクトリの種類	読み込み許可 (r)	書き込み許可 (w)	実行許可 (x)	スティッキービット (t)
一時ファイル用のディレクトリ	○	○	○	△
スプール用のディレクトリ	○	○	○	△
システム実行ログ用のディレクトリ	○	○	○	×
トレース用のディレクトリ	○	○	○	△

(凡例)

- ：設定が必須です。
- △：システムの運用方針に従って設定します。
- ×：設定しません。

ディレクトリのスティッキービットは、システムの運用方針に従い、設定します。

ディレクトリにスティッキービットの設定がない場合、ディレクトリに書き込み許可があれば、ディレクトリ直下の任意のファイルを削除できます。

ディレクトリにスティッキービットを設定した場合、ディレクトリに書き込み許可があっても、ディレクトリの所有者、またはファイルの所有者の場合だけ、ディレクトリ直下の任意のファイルを削除できます。

(2) ファイルシステム

スプールなどは、業務によって大容量となる場合があるため、専用のファイルシステムを作成して使用することをお勧めします。

2.1.4 UNIX 互換コマンドを使用するための設定をする

(1) 既存のジョブ定義スクリプトで実行ファイル形式の UNIX 互換コマンドを使用するための定義

既存のジョブ定義スクリプトで実行ファイル形式の UNIX 互換コマンドを使用する場合は、PATH 環境変数に UNIX 互換コマンドがインストールされているディレクトリへのパスを設定することで、ジョブ定義スクリプトの修正が不要になります。このとき、環境ファイルの export パラメーターを使用して PATH 環境変数の値の先頭にパスを設定することで、すでに UNIX 互換コマンドと同名のコマンドが存在していても、JP1/Advanced Shell のジョブ定義スクリプトでは必ず UNIX 互換コマンドを動作させることができます。

ジョブ定義スクリプトを実行する各環境で、正しいパスが設定されていることを確認してからジョブ定義スクリプトを実行してください。

(2) スクリプト形式の UNIX 互換コマンドを使うための準備 (Windows の場合)

スクリプト形式の UNIX 互換コマンドは、JP1/Advanced Shell が提供するサンプルスクリプトファイルを使用します。

スクリプト形式の UNIX 互換コマンド (chmod コマンドと su コマンドなど) は、JP1/Advanced Shell が提供するサンプルスクリプトファイルを基に、次の手順で実行します。

1. 次の場所にあるサンプルスクリプトファイルのうち、使用するファイルを任意のフォルダにコピーする。

インストール先フォルダ¥JP1ASE¥sample

2. コピーしたファイルの名称を各コマンド名にリネームする。

例えば、サンプルスクリプトファイル「script_chmod1」の場合はファイル名を「chmod」へ、サンプルスクリプトファイル「script_su1」の場合はファイル名を「su」へリネームします。何もしないコマンドとしたい場合は、サンプルスクリプトファイル「script_0」をコピーしてリネームします。

3. サンプルスクリプトを絶対パスや相対パス指定でなく、ファイル名だけ指定して実行したい場合は、次のどちらかを実行する。

- 環境変数 PATH で定義されたフォルダに、実行するサンプルスクリプトを格納する
- 環境変数 PATH に、実行するサンプルスクリプトを格納したフォルダのパスを追加する

4. 必要に応じて、KNAX6831-I メッセージの出力抑止を定義する。

サンプルスクリプトの実行後に KNAX6831-I メッセージが出力させたくない場合は、ジョブ環境ファイルに次の内容を記述してください。

```
#-adsh_conf JOBLLOG_SUPPRESS_MSG      KNAX6831-I
```

システム内のすべてのジョブ定義スクリプトに対して KNAX6831-I メッセージの出力を抑止したい場合は、システム環境ファイルに記述してください。

5. ジョブ定義スクリプトを実行する。

手順 4. で作成したジョブ環境ファイルを使用して、ジョブ定義スクリプトを実行してください。システム環境ファイルに指定した場合は、手順 4. で指定した内容は自動的に読み込まれます。

2.1.5 JP1/AJS からジョブを起動するためのシェルを設定する (Linux の場合)

JP1/AJS からジョブを起動する場合に使用するログインシェルを次に示します。該当するログインシェルが使用できるよう設定してください。

OSの種類	ログインシェル
Linux	bash

注意

JP1/AJS からジョブを起動する場合、adshexec コマンドがログインシェルの子プロセスとして動作すると、強制終了などが発生した際に、adshexec コマンドのジョブ実行結果が JP1/AJS に渡る前にログインシェルのプロセスが終了し、JP1/AJS - View 上にジョブ実行結果が反映されないことがあります。この現象を防ぐには、ログインシェルのプロセスを上書きしてから起動する処理になっているか、ログインスクリプトファイルの定義を確認してください (trap コマンドの指定を削除するなど)。

定義の詳細については、マニュアル「JP1/Automatic Job Management System 3 構築ガイド 1」またはマニュアル「JP1/Automatic Job Management System 3 トラブルシューティング」を参照してください。

2.1.6 子孫ジョブとして起動するファイルを定義する

ジョブ定義スクリプト中にコマンド名としてほかのジョブ定義スクリプトを指定できます。これによって、adshexec コマンドに指定したジョブ定義スクリプトを JP1/Advanced Shell のジョブとして実行できます。次のような場合に有効です。

- ユーザーの既存資産のシェルスクリプトを UNIX 環境から Windows 環境へ移行する場合
- UNIX 環境で OS のシェルで動作していた既存のシェルスクリプトを、内容を書き換えしないで JP1/Advanced Shell のジョブとして実行する場合

子孫プロセスとして実行されるジョブ定義スクリプトのうち、特定の環境設定パラメーターによって実行されたジョブを子孫ジョブと呼びます。また、JP1/AJS やログインシェルなどから実行するジョブのうち、子孫ジョブ以外のジョブをルートジョブと呼びます。

ジョブ定義スクリプトファイルの子孫ジョブとして起動させる場合、動作させるファイルの条件を環境ファイルに設定しておきます。環境設定パラメーターの概要を次に示します。

- CHILDJOB_EXT パラメーター
子孫ジョブとして実行するジョブ定義スクリプトファイルの拡張子を定義する
- CHILDJOB_PGM パラメーター
子孫ジョブとして実行するよう読み替えるパスを定義する
- CHILDJOB_SHEBANG パラメーター
子孫ジョブとして実行するジョブ定義スクリプトファイルの実行プログラムパスを定義する

また、CHILDJOB_SHEBANG パラメーターのデフォルト定義に合致するジョブ定義スクリプトファイルを作成すれば、子孫ジョブとして実行されます。

注意事項

ルートジョブと子孫ジョブを同じ環境ファイルパラメーターで動作させたい場合は、ジョブの実行中に環境変数 ADSH_ENV の値や環境ファイルの内容を変更しないでください。

2.1.7 ジョブの種類ごとのジョブ実行ログの出力内容

ジョブ実行ログの出力内容は、実行したジョブの種類によって次のように異なります。

(1) ルートジョブ実行時のジョブ実行ログの出力先と出力内容

ルートジョブ実行時のジョブ実行ログの出力先と出力内容について、拡張出力モードおよび最小出力モード（OUTPUT_MODE_ROOT パラメーターで指定）に分けて説明します。

(a) 拡張出力モードを選択した場合

拡張出力モードを選択した場合のジョブ実行ログの出力先を次に示します。

メッセージの出力先	内容
JOBLOG	スプール内のファイルに出力されます。
スクリプトイメージ	スプール内のファイルに出力されます。
標準出力の出力先	次のどちらかで指定した出力先へ出力されます。 <ul style="list-style-type: none">• adshexec コマンドの -s オプション• 環境ファイルの OUTPUT_STDOUT パラメーター
標準エラー出力の出力先	スプール内のファイルに出力されます。

ジョブごとにスプールジョブディレクトリが作成されます。

ジョブ実行後、標準出力を除いたジョブ実行ログの内容が標準エラー出力に出力されます。

(b) 最小出力モードを選択した場合

最小出力モードを選択した場合のジョブ実行ログの出力先を次に示します。

メッセージの出力先	内容
JOBLOG	スプール内のファイルに、出力抑止対象外のメッセージが出力されます。
スクリプトイメージ	最小出力モードでは、一部のメッセージの出力が抑止されます。抑止されるメッセージの詳細は、マニュアル「JP1/Advanced Shell」の「ジョブの種類ごとのジョブ実行ログの出力内容」の説明を参照してください。
標準出力の出力先	スプール内のファイルには出力されません。プロセス起動時の出力先に出力されます。
標準エラー出力の出力先	標準エラー出力と標準出力の出力抑止対象外のメッセージが出力されます。また、JOBLOG の出力抑止対象外のメッセージが標準エラー出力に出力されます。

なお、ジョブ終了時にジョブ実行ログは標準エラー出力に出力されません。

(2) 子孫ジョブ実行時のジョブ実行ログの出力先と出力内容

子孫ジョブ実行時のジョブ実行ログの出力先と出力内容について、拡張出力モード、および最小出力モード（OUTPUT_MODE_CHILD パラメーターで指定）に分けて説明します。子孫ジョブのスプールジョ

ブをルートジョブのプールジョブへマージする場合の出力内容を示します。子孫ジョブのプールジョブを削除する場合の出力内容は、マニュアル「JP1/Advanced Shell」の「ジョブの種類ごとのジョブ実行ログの出力内容」の説明を参照してください。

(a) 拡張出力モードを選択した場合

拡張出力モードを選択した場合のジョブ実行ログの出力先を次に示します。

メッセージの出力先	内容
JOBLOG	一時的に子孫ジョブのプール内のファイルに出力されます。 子孫ジョブ終了時に、ルートジョブのJOBLOGに子孫ジョブのJOBLOGがマージされます。子孫ジョブのJOBLOGの前後に、JOBLOGの出力開始を示す記号「>>>>>> [JOBLOG] パス名」と、出力終了を示す記号「<<<<<< [JOBLOG] パス名」が出力されます。
スクリプトイメージ	一時的に子孫ジョブのプール内のファイルに出力されます。 子孫ジョブ終了時に、ルートジョブのスクリプトイメージに子孫ジョブのスクリプトイメージがマージされます。
標準出力の出力先	プロセス起動時の出力先に出力されます。
標準エラー出力の出力先	プロセス起動時の出力先に出力されます。 子孫ジョブの標準エラー出力の前後に、標準エラー出力の出力開始を示す記号「>>>>>> [STDERR] パス名」と、出力終了を示す記号「<<<<<< [STDERR] パス名」が出力されます。

ジョブ実行中に子孫ジョブのプールジョブディレクトリが作成されますが、ジョブ実行後に削除されます。

また、次に示すヘッダ行はマージされません。

```

-----
Advanced Shell バージョン番号

[ジョブ情報]
ジョブ識別子          : ジョブ識別子
プールジョブディレクトリパス : スプールジョブディレクトリパス
実行日付              : 実行日付
システム環境ファイルパス : 環境ファイルパス (システム環境ファイル)
ジョブ環境ファイルパス  : 環境ファイルパス (ジョブ環境ファイル)
ホスト名              : ホスト名
[Automatic Job Management Systemから渡された環境変数]
JP1/AJSから渡された環境変数
-----
***** ジョブコントローラのメッセージ出力 *****

```

(b) 最小出力モードを選択した場合

最小出力モードを選択した場合のジョブ実行ログの出力先を次に示します。

メッセージの出力先	内容
JOBLOG	<p>一時的に子孫ジョブのスプール内のファイルに出力抑止対象外のメッセージが出力されます。メッセージが出力されている場合、子孫ジョブのJOBLOGはルートジョブのJOBLOGにマージされ、子孫ジョブのJOBLOGの前後に、JOBLOGの出力開始を示す記号「>>>>> [JOBLOG] パス名」と、出力終了を示す記号「<<<<<< [JOBLOG] パス名」が出力されます。</p> <p>メッセージが出力されていない場合、子孫ジョブのJOBLOGはルートジョブのJOBLOGにマージされません。</p> <p>最小出力モードでは、一部のメッセージの出力が抑止されます。抑止されるメッセージの詳細は、マニュアル「JP1/Advanced Shell」の「ジョブの種類ごとのジョブ実行ログの出力内容」の説明を参照してください。</p>
スクリプトイメージ	一時的にスプール内のファイルに出力されますが、ルートジョブのスクリプトイメージにはマージされません。
標準出力の出力先	<p>プロセス起動時の出力先に出力されます。</p> <p>標準出力の出力抑止対象外のメッセージが出力されます。</p> <p>出力抑止したメッセージは出力されません。</p>
標準エラー出力の出力先	<p>プロセス起動時の出力先に出力されます。</p> <p>標準エラー出力の出力抑止対象外のメッセージが出力されます。また、JOBLOGの出力抑止対象外のメッセージが出力されます。</p> <p>出力抑止したメッセージは出力されません。</p>

ジョブ実行中に子孫ジョブのスプールジョブディレクトリが作成されますが、ジョブ実行後に削除されます。

2.1.8 ジョブ実行結果とログの出力情報を定義する

ジョブの実行結果はスプールディレクトリに出力され、出力内容の一部はジョブ実行ログとして参照できます。また、トラブル発生時にログを採取し、トラブルの原因を調査できます。これらのログの出力先や出力内容を環境ファイルで定義します。

JP1/Advanced Shell を運用しているときに出力されるログ情報と格納先について、次の表に示します。

表 2-5 JP1/Advanced Shell 運用時に出力されるログ情報と格納先

ログ情報	出力される情報	格納先
ジョブ実行ログ	バッチジョブのログ	スプूलルートディレクトリの配下
システム実行ログ	JP1/Advanced Shell の統括的な実行ログ	環境ファイルの LOG_DIR パラメーター※で指定したディレクトリ

注※

パラメーターの指定がない場合、デフォルト値が適用されます。

スプूलの出力情報と、それぞれのログの出力情報の定義方法を次に説明します。

(1) スプールの出力情報を定義する

スプールに関するパラメーターについて、定義する出力情報ごとに説明します。

(a) スプールジョブ作成抑止機能の使用有無を決定する

スプールジョブ作成抑止機能を使用すると、スプールジョブが作成されなくなるため、スプールディレクトリのディスク容量の単調増加を防げます。また、スプールディレクトリ下の不要なディレクトリやファイルを削除する必要がなくなります。

スプールジョブ作成抑止機能の使用は SPOOLJOB_CREATE パラメーターで設定します。詳細については、マニュアル「JP1/Advanced Shell」の「SPOOLJOB_CREATE パラメーター」の説明を参照してください。

スプールジョブ作成抑止機能を使用している間は、ルートジョブおよび子孫ジョブは常に最小出力モードで動作します。

また、スプールジョブ作成抑止機能を使用した場合でも、スプールディレクトリは必要です。

(b) スプールルートディレクトリのパス名を定義する

スプールルートディレクトリのパス名を定義するパラメーターを次に示します。

- SPOOL_DIR パラメーター：スプールルートディレクトリのパス名を定義します。

(c) スプールジョブディレクトリまたはファイルのパーミッションを変更する【UNIX 限定】

ジョブが終了すると、実行結果はジョブごとに作成されるスプールジョブディレクトリへ出力されます。このときディレクトリまたはその下のファイルに設定されるパーミッションは、次のパラメーターで変更できます。

- PERMISSION_SPOOLJOB_DIR パラメーター
スプールジョブディレクトリのパーミッションを変更したい場合に指定します。
デフォルトは 700 です。
- PERMISSION_SPOOLJOB_FILE パラメーター
スプールジョブディレクトリ下のファイルのパーミッションを変更したい場合に指定します。
デフォルトは 600 (.DBG ファイルは 666) です。

(d) スプールジョブの標準出力と標準エラー出力の出力内容を定義する

ジョブを実行すると、ジョブの出力結果のほかに、JP1/Advanced Shell の情報メッセージ、警告メッセージ、およびジョブ実行ログが出力されます。また、標準出力と標準エラー出力はスプールジョブディレクトリ下のファイルへ出力されます。

ジョブの実行結果だけをほかのプログラムで利用するなどの理由で、標準出力と標準エラー出力のプールジョブディレクトリ下のファイルへの出力を抑止したい場合には、次のパラメーターまたはコマンドのオプションで最小出力モードを指定します。

- OUTPUT_MODE_ROOT パラメーター
ルートジョブに対して、拡張出力モード、または最小出力モードを指定します。
- OUTPUT_MODE_CHILD パラメーター
子孫ジョブに対して、拡張出力モード、または最小出力モードを指定します。
- adshexec コマンドの-m オプション
ジョブに対して、拡張出力モード、または最小出力モードを指定します。
- adshscripttool コマンドの-m オプション
子孫ジョブに対して、最小出力モードを指定します。

これらのパラメーターとオプションを指定しなかった場合は拡張出力モードとなり、標準出力と標準エラー出力はプールジョブディレクトリ下のファイルへ出力されます。

なお、最小出力モードでは、JP1/Advanced Shell の情報メッセージと警告メッセージは標準出力、標準エラー出力へ出力しません。また、ジョブ終了時にジョブ実行ログを標準エラー出力へ出力しません。さらに、最小出力モードでは、出力抑止したメッセージはプールジョブディレクトリ下のジョブ実行ログにも出力しません。

拡張出力モードおよび最小出力モードの出力内容の差異については、マニュアル「JP1/Advanced Shell」の「ジョブ実行ログへの情報メッセージと警告メッセージの出力を抑止する」の説明を参照してください。

(2) ジョブ実行ログの出力情報を定義する

ジョブ実行ログに関して環境設定時に設定する内容について説明します。ジョブ実行ログの出力内容については「[4.2.1 ジョブ実行ログ](#)」を参照してください。

(a) 標準エラー出力に出力するジョブ実行ログの種類を定義する

ジョブが終了すると、ジョブ実行ログとして次に示す内容が標準エラー出力へ出力されます。出力されたジョブ実行ログは、adshexec コマンドの実行時の端末画面や、JP1/AJS - View の「実行結果詳細」ダイアログボックスなどに表示されます。

- JOBLOG ファイル（コマンドの実行結果やファイルの割り当て結果など、ジョブの動作状況を示すメッセージ）
- ジョブ定義スクリプト
- ジョブ実行中の標準エラー出力の内容

このうち、ジョブ実行中の標準エラー出力の内容だけを標準エラー出力へ出力させるには、次のパラメーターを指定します。これによって、ジョブ実行ログの出力内容を限定できます。

- JOBEXECLOG_PRINT パラメーター

ジョブを最小出力モードで実行した場合は、JOBEXECLOG_PRINT パラメーターの指定に関係なく、ジョブ終了時にジョブ実行ログを標準エラー出力に出力しません。

(3) システム実行ログの出力情報を定義する

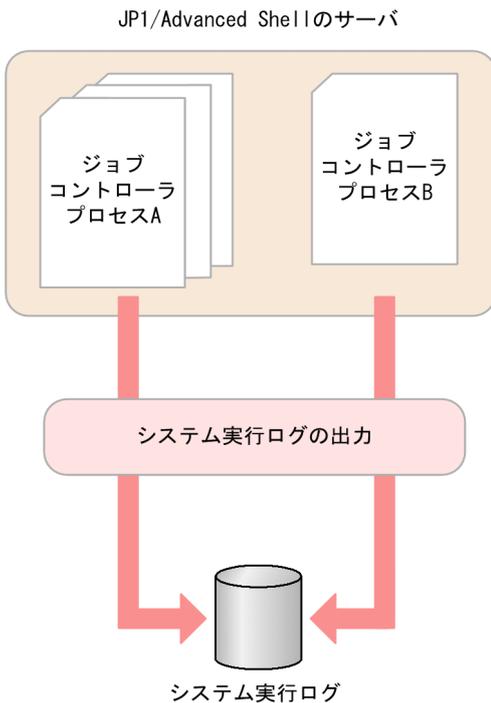
システム実行ログとは、バッチジョブの実行履歴を統括的に参照するためのシステム管理者向けのログ情報です。

このログ情報は、環境ファイルの LOG_DIR パラメーターで指定したディレクトリ下の AdshLog.log に出力されます。パラメーターに設定した条件（サイズなど）に従って、ファイル名のローテーション（AdshLog_1.log, AdshLog_2.log, …AdshLog_N.log）が実施されます。システム実行ログのファイルはローテーション時に新しく作成されるため、ファイルの所有者はローテーション時のユーザーになります。

(a) システム実行ログの出力の流れ

システム実行ログには、各ジョブコントローラプロセスで実行しているバッチジョブの情報が出力されます。環境ファイルにシステム実行ログの出力先、サイズおよび面数を指定できます。システム実行ログの出力の流れを次に示します。

図 2-1 システム実行ログの出力の流れ



システム実行ログは次のように作成されます。

- システム実行ログに出力するメッセージを集め、CSV形式で出力します。
出力するメッセージについては、マニュアル「JP1/Advanced Shell」の「メッセージの出力先」の説明を参照してください。

- ローテーションを行い、バックアップが作成されます。
 - 環境ファイルの LOG_FILE_SIZE パラメーターに指定されたファイルサイズを超える直前に、システム実行ログのファイル名を変更してバックアップを作成し、新たにシステム実行ログを作成して出力を継続します。
 - バックアップのファイル名は、AdshLog_N.log (Nは整数) となります。Nには新しいバックアップから昇順に1から番号を割り当てます。
 - 環境ファイルの LOG_FILE_CNT パラメーターに指定された面数のバックアップを作成し、面数を超えた場合は古いバックアップを削除します。

(b) システム実行ログを出力するために必要なパラメーター

システム実行ログを出力するために必要なパラメーターを次に示します。

- LOG_DIR パラメーター
システム実行ログを出力するディレクトリのパス名を定義します。
- LOG_FILE_CNT パラメーター
システム実行ログをバックアップする面数を定義します。
- LOG_FILE_SIZE パラメーター
システム実行ログを出力するファイルサイズを定義します。

複数のユーザーが同じファイルにシステム実行ログを出力している場合には、LOG_FILE_CNT と LOG_FILE_SIZE は、最後にシステム実行ログの出力を開始したユーザーの設定値が有効になります。このため、LOG_FILE_CNT と LOG_FILE_SIZE は同じ値で運用することを推奨します。

(c) システム実行ログの出力内容

システム実行ログに出力されるメッセージの例を次に示します。

```
seqnum=1, date=2013-12-06T18:27:12.045+09:00, pgmid=adshexec, jobid=70, pid=6616,
msgid=KNAX0004-I, msg="ジョブ識別子=000070, JP1NBQQueueName=, ジョブ番号="
seqnum=2, date=2013-12-06T18:27:12.060+09:00, pgmid=adshexec, jobid=70, pid=6616,
msgid=KNAX0091-I, msg="JOB1 ジョブが開始しました。"
seqnum=3, date=2013-12-06T18:27:12.060+09:00, pgmid=adshexec, jobid=70, pid=6616,
msgid=KNAX7901-I, msg="ジョブコントローラは、ジョブ終了時にすべての非同期実行プロセスの完了を待ちます。"
seqnum=4, date=2013-12-06T18:27:12.060+09:00, pgmid=adshexec, jobid=70, pid=6616,
msgid=KNAX7902-I, msg="ジョブコントローラは、""端末入力モード""で動作します。"
```

システム実行ログで、メッセージテキストの前に付加されるデータの意味を次に示します。

システム実行ログに出力されるデータ	データの意味
seqnum	メッセージの通し番号
date	出力した日時 (yyyy-mm-ddThh:mm:ss.sssTZD の形式)

システム実行ログに出力されるデータ	データの意味
pgmid	プログラム ID ジョブコントローラの場合、adshexec が出力されます。
jobid	ジョブ識別子
pid	プロセス ID
msgid	出力メッセージのメッセージ ID
msg	出力メッセージのメッセージテキスト

2.1.9 スクリプト拡張コマンドの終了コードを定義する

スクリプト拡張コマンドが失敗および成功した場合の終了コードを標準の値から変更したいときに指定します。

- ADSHCMD_RC_ERROR パラメーター：スクリプト拡張コマンドが失敗した場合の終了コードを定義します。
- ADSHCMD_RC_SUCCESS パラメーター：スクリプト拡張コマンドが成功した場合の終了コードを定義します。

詳細については、マニュアル「JP1/Advanced Shell」の「スクリプト拡張コマンドの終了コードとエラー発生時の動作」の説明を参照してください。

2.1.10 ジョブ強制終了時にユーザー固有の後処理を実行する

JP1/Advanced Shell では、JP1/AJS からの強制終了、UNIX 版での SIGTERM シグナルによる強制終了、および Windows での taskkill コマンドによる強制終了 (TerminateProcess などによるプロセス即時終了) を受けたときにユーザー固有の後処理を実行できます。これによって、強制終了要求を受けたことを契機として独自の終了処理を実行するなど、柔軟な運用ができます。強制終了要求を受けたときにユーザー固有の後処理を実行する場合は、環境設定パラメーター TRAP_ACTION_SIGTERM を定義してください。

なお、環境設定パラメーター TRAP_ACTION_SIGTERM は UNIX 版と Windows 版で指定できるオペラントが異なります。環境設定パラメーター TRAP_ACTION_SIGTERM については、マニュアル「JP1/Advanced Shell」の「TRAP_ACTION_SIGTERM パラメーター」の説明を参照してください。

使用例を次に示します。

環境ファイルの内容

```
#-adsh_conf TRAP_ACTION_SIGTERM TERM
```

ジョブ定義スクリプトの内容

```
#-adsh_job JOB01
trap "UAP_TERM" TERM
UAP01
```

→UAP01 実行中に強制終了要求を受けた場合、ジョブコントローラは「UAP_TERM」というユーザープログラムを実行したあと、子孫プロセスの強制終了といった後処理を実行してからジョブを終了します。

2.2 環境変数を設定する

JP1/Advanced Shell で使用できる環境変数を次の表に示します。

注意事項

JP1/Advanced Shell では、名称が ADSH から始まるシェル変数・環境変数を設定・参照しています。このため、このマニュアルに記載されている使用目的以外では、名称が ADSH から始まるシェル変数・環境変数は使用しないでください。

表 2-6 JP1/Advanced Shell で使用できる環境変数

環境変数名	設定する内容	値が自動的に設定される場合の設定時期	値の設定可否
ADSH_ENV	ジョブ環境ファイルのファイル名。	カスタムジョブで起動した場合のジョブ開始時	可※
ADSH_JOBRC_FATAL	構文エラーなど、ジョブを続行できない致命的なエラーが発生した場合のジョブの終了コード。 設定方法については「(1) 環境変数 ADSH_JOBRC_FATAL (ジョブ続行不可エラー発生時の終了コードを設定する)」を参照してください。	(自動的に設定されない)	可※
AJS_BJEX_STOP	JP1/AJS からの強制終了で使用するインターフェース。PC ジョブや UNIX ジョブで JP1/Advanced Shell のバッチジョブを定義する場合は、この環境変数を定義する必要がある。OS の設定で定義するのではなく、PC ジョブや UNIX ジョブの定義で定義すること。	カスタムジョブで起動した場合のジョブ開始時	可 (TERM だけ設定可能)

注※

ジョブ定義スクリプトファイルや環境ファイルで設定した場合、その指定は、ジョブ定義スクリプトから起動する子孫ジョブやルートジョブにだけ有効となります。

環境変数の詳細については、マニュアル「JP1/Advanced Shell」の「環境変数を設定する」の説明を参照してください。

2.2.1 ジョブを続行できないエラーが発生したときの終了コードを定義する

メモリ不足やジョブ定義スクリプトの解析エラーなど、ジョブの実行を継続できないエラーによってジョブが終了した場合、ジョブコントローラの終了コードは 1 となります。しかし、環境変数 ADSH_JOBRC_FATAL に値を設定しておくことで、この場合の終了コードを 1~255 に変更できます。

環境変数 ADSH_JOBRC_FATAL の設定方法および適用可否の詳細については、マニュアル「JP1/Advanced Shell」の「環境変数 ADSH_JOBRC_FATAL」の説明を参照してください。

(1) 環境変数 ADSSH_JOBRC_FATAL (ジョブ続行不可エラー発生時の終了コードを設定する)

ジョブが続行できなくなってエラー終了した場合のジョブコントローラの終了コードを設定します。設定した終了コードは、adshexec コマンドから実行されたジョブに対して適用されます。

システム全体でこの環境変数の設定値を有効にする方法を次に示します。

- Windows の場合
システム環境変数として環境変数 ADSSH_JOBRC_FATAL を定義する。
- UNIX 版の場合
/etc/profile に環境変数 ADSSH_JOBRC_FATAL の設定処理を記述する。

この環境変数を設定しなかった場合、ジョブが続行できないエラーで終了した際のジョブコントローラの終了コードは 1 となります。

(a) 環境変数に設定できる値

終了コード ~<符号なし整数>((1~255))

ジョブ続行不可時の終了コードを指定します。前の桁を 0 で埋めて「001」のように指定した場合は、上位のゼロを削除して「1」と扱います。

(b) 注意事項

- 環境ファイルの export パラメーターを使用して環境変数 ADSSH_JOBRC_FATAL を定義した場合や、シェル変数 ENV に指定したファイル内およびジョブ定義スクリプト内で環境変数 ADSSH_JOBRC_FATAL の定義・変更をした場合、そのジョブ内ではこの機能は有効になりません。ただし、そのジョブから起動した別のジョブで有効になります。
- 環境変数 ADSSH_JOBRC_FATAL はジョブの最終的な終了コードを定義します。各コマンドやジョブステップの終了コードには影響しません。
- 次の値が設定された場合、ジョブは実行されないで終了コード 255 でエラー終了します。
 - 4 文字以上の値 (例: 1234)
 - 指定範囲外の値 (例: 500)
 - 数値以外の文字 (例: 1A4, +8, 8.0)
 - 0 文字の値 (空文字列)
- エラー発生時に環境変数 ADSSH_JOBRC_FATAL を適用するかどうかの判定は、ジョブごとに独立しています。そのため、ルートジョブや子孫ジョブの中だけでジョブの実行を継続できないエラーが発生した場合に、ほかのルートジョブや子孫ジョブに対して環境変数 ADSSH_JOBRC_FATAL が適用されて終了コードが変更されることはありません。

3

JP1/AJS のジョブ定義

ここでは、JP1/AJS でジョブを定義する手順について説明します。

3.1 JP1/AJS でジョブを定義する

JP1/AJS を使用する場合の環境情報の設定について説明します。

JP1/AJS でジョブの実行を自動化する場合、JP1/AJS - View でジョブを登録します。JP1/AJS - View では、運用に使用するコマンド、バッチファイルなどをジョブとして定義して、それぞれのジョブの実行順序を関連づけることで、システム運用を自動化します。

JP1/AJS - View でジョブを定義する方法には、次の種類があります。

- カスタムジョブ
- PC ジョブ (Windows の場合)
- UNIX ジョブ (UNIX の場合)

基本ガイドでは、これらのうち、PC ジョブまたは UNIX ジョブでバッチジョブを定義する手順を説明します。

JP1/AJS - View の詳細については、マニュアル「JP1/Automatic Job Management System 3 操作ガイド」を参照してください。

3.1.1 ジョブネットを定義して実行する

JP1/AJS でジョブの実行を自動化する場合、登録したカスタムジョブ、PC ジョブ (Windows の場合)、または UNIX ジョブ (UNIX の場合) を JP1/AJS - View でジョブネットに定義し、ジョブネットを実行します。JP1/AJS - View の詳細については、マニュアル「JP1/Automatic Job Management System 3 操作ガイド」のジョブ定義の説明を参照してください。

JP1/AJS3 - View でのジョブネットを定義して実行する手順を次に示します。

1. Windows の [スタート] メニューから、[すべてのプログラム] - [JP1_Automatic Job Management System 3 - View] - [ジョブシステム運用] を選択する。
[ログイン] 画面が表示されます。
2. ユーザー名、パスワード、および接続ホスト名を指定してログインする。
[JP1/AJS3 - View] ウィンドウが表示されます。
3. [編集] - [新規作成] - [ジョブネット] を選択する。
[詳細定義] - [ジョブネット] ダイアログボックスが表示されます。
4. ジョブネットに属性などを定義して、[OK] ボタンをクリックする。
運用環境に応じて、[実行エージェント] に適切な内容を指定してください。省略することもできます。
JP1/AJS の項目については、JP1/AJS のマニュアルを参照してください。

詳細定義 - [ジョブネット]

ユニット名: jolas_job

コメント:

実行エージェント: hostname

定義 | 属性

多重起動: 不可能 可能

保存世代数: 1

優先順位: なし

打ち切り時間: システム設定に従う

スケジューリング方式: スケジュールスキップ 多重スケジュール

ジョブネット監視: 実行所要時間: 分

実行順序制御: する しない

接続範囲: 同一サービス 別サービス

接続ホスト名:

接続サービス名:

ジョブネットコネクタ名:

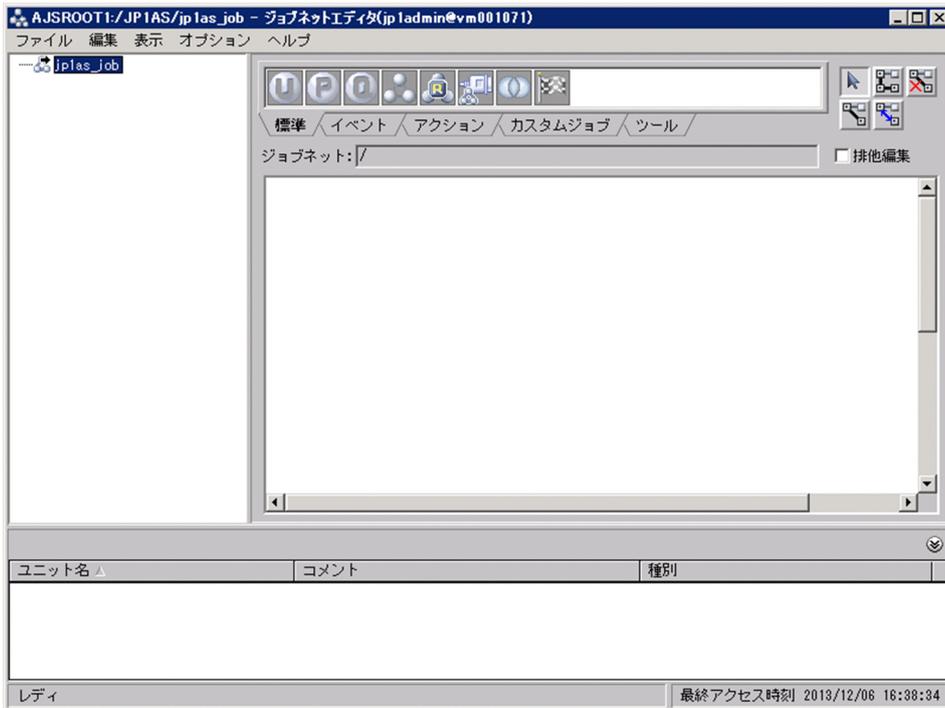
実行順序制御方式: 同期 非同期

OK キャンセル ヘルプ

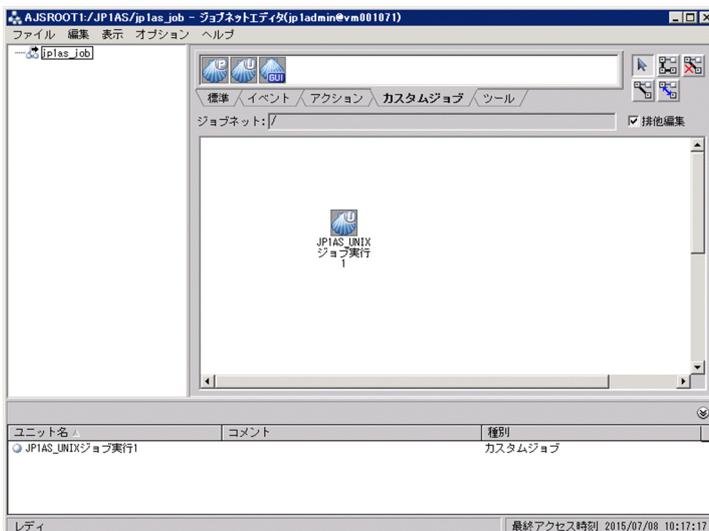
これによってジョブネットが作成され、リストエリアに表示されます。

5. 作成したジョブネットをダブルクリックする。

[ジョブネットエディタ] ウィンドウが表示されます。



6. ジョブを定義したり関連づけたりするときにほかのユーザーがアクセスできないように、[排他編集] をチェックする。
7. アイコンリストから必要なカスタムジョブ、PC ジョブ、または UNIX ジョブのアイコンをマップエリアへドラッグする。



[詳細定義- [Custom Job]] ダイアログボックス、[詳細定義- [PC Job]] ダイアログボックス、または [詳細定義- [UNIX Job]] ダイアログボックスが表示されます。

以降の手順は、PC ジョブまたは UNIX ジョブのどちらを使用するかによって手順が異なります。

PC ジョブを使用する場合は「3.1.2 PC ジョブでバッチジョブを定義する」を参照してください。

UNIX ジョブを使用する場合は、「3.1.3 UNIX ジョブでバッチジョブを定義する」を参照してください。

(1) ジョブネットの定義内容に関する注意事項

- JP1/AJS から JP1/Advanced Shell のジョブコントローラを起動したときの実行時ディレクトリ
JP1/AJS から JP1/Advanced Shell のジョブコントローラを起動した場合の実行時ディレクトリは、JP1/AJS - Agent (または JP1/AJS - Manager) が JP1/Advanced Shell のジョブコントローラを起動するときの実行時ディレクトリになります。JP1/AJS - Agent (または JP1/AJS - Manager) が JP1/Advanced Shell のジョブコントローラを起動するときの実行時ディレクトリについては、マニュアル「JP1/Automatic Job Management System 3 構築ガイド 1」を参照してください。実行時ディレクトリは、JP1/AJS のマニュアルでは、ワークパス (ワークディレクトリ) と表記されています。
- JP1/AJS から JP1/Advanced Shell のジョブコントローラを起動したときの環境変数
JP1/AJS から Windows の JP1/Advanced Shell のジョブコントローラを起動した場合、通常 JP1/AJS サービス起動時にはシステム環境変数の設定を有効にし、ユーザー環境変数は読み込まれません。詳細は JP1/AJS のマニュアルを参照してください。
- 言語種別を英語に設定した海外版の JP1/AJS - Manager と接続して使用する
言語種別を英語に設定した海外版の JP1/AJS - Manager と接続してジョブを定義する場合、[実行定義] ダイアログボックスのジョブの定義情報には英数字 (ASCII) だけを使用してください。

3.1.2 PC ジョブでバッチジョブを定義する

PC ジョブで JP1/Advanced Shell のバッチジョブを定義する場合に必要な項目について示します。

(1) バッチジョブを定義する場合

- 実行ファイル名
[詳細定義 - [PC Job]] ダイアログボックスの [定義] タブの「実行ファイル名」、またはユニット定義ファイルの「sc="スクリプトファイル名"」に、adshexec コマンドのパスを指定します。

```
"インストール先フォルダ¥JP1ASE¥bin¥adshexec.exe"
```
- パラメーター
[詳細定義 - [PC Job]] ダイアログボックスの [定義] タブの「パラメーター」、またはユニット定義ファイルの「prm="パラメーター"」に、adshexec コマンドのオプション、ジョブ定義スクリプトファイル名、および実行時パラメーターを指定します。
- 環境変数
[詳細定義 - [PC Job]] ダイアログボックスの [定義] タブの「環境変数」、またはユニット定義ファイルの「env="環境変数"」に、次の内容を指定します。

```
AJS_BJEX_STOP=TERM
```

JP1/Advanced Shell のバッチジョブとしての指定例を次の図に示します。

図 3-1 [詳細定義－ [PC Job]] ダイアログボックスの [定義] タブの指定例

3.1.3 UNIX ジョブでバッチジョブを定義する

UNIX ジョブで JP1/Advanced Shell のバッチジョブを定義する場合に必要な項目について示します。

- スクリプトファイル名

[詳細定義- [UNIX Job]] ダイアログボックスの [定義] タブの「スクリプトファイル名」, またはユニット定義ファイルの「sc="スクリプトファイル名"」に, adshexec コマンドのパスを指定します。

```
/opt/jp1as/bin/adshexec
```

また, 1 行目に#!で開始する adshexec コマンドのパス (例: #!/opt/jp1as/bin/adshexec) を記述し, 実行権限を付与したジョブ定義スクリプトファイルのパスを指定することもできます。

```
ジョブ定義スクリプトファイルのパス
```

- コマンド文

[詳細定義- [UNIX Job]] ダイアログボックスの [定義] タブの「コマンド文」、またはユニット定義ファイルの「te="コマンドテキスト"」に、コマンドテキストの一部として「スクリプトファイル名」と同様の方法で adshexec コマンドのパス、またはジョブ定義スクリプトファイルのパスを指定できます。コマンド文で定義したジョブを JP1/AJS で強制終了した場合、次の制限があります。

- 強制終了の操作をするタイミングによっては、JP1/AJS-View からジョブ実行ログやジョブの標準エラー出力の内容を参照できないことがあります。この場合、ジョブ実行ログについては、スプールジョブディレクトリを参照することで内容を確認できます。
 - ジョブ実行ログに出力されるジョブの終了コードは 143 ですが、JP1/AJS - View から参照できるジョブの終了コードは-1 になります。
- パラメーター

[詳細定義- [UNIX Job]] ダイアログボックスの [定義] タブの「パラメーター」、またはユニット定義ファイルの「prm="パラメーター"」に、adshexec コマンドのオプション、ジョブ定義スクリプトファイル名、および実行時パラメーターを指定します。

スクリプトファイル名にジョブ定義スクリプトファイル名を指定した場合は、実行時パラメーターだけを指定します。

- 環境変数

[詳細定義- [UNIX Job]] ダイアログボックスの [定義] タブの「環境変数」、またはユニット定義ファイルの「env="環境変数"」に、次の内容を指定します。

```
AJS_BJEX_STOP=TERM
```

JP1/Advanced Shell のバッチジョブとしての指定例を次の図に示します。この例では、「スクリプトファイル名」に adshexec コマンド、およびジョブ定義スクリプトファイルパスを指定しています。

図 3-2 「詳細定義- [UNIX Job]」 ダイアログボックスの [定義] タブの指定例 (adshexec コマンドを指定する場合)

ユニット名	sample
コメント	
実行エージェント	
定義 転送ファイル 属性	
コマンド文	
スクリプトファイル名	/opt/jp1as/bin/adshexec
パラメーター	/home/user1/scripts/sct02.ash param1 param2
環境変数	AJS_BJEX_STOP=TERM
環境変数ファイル名	
ワークパス	
実行優先順位	なし
標準入力ファイル名	
標準出力ファイル名	<input type="checkbox"/> 追加書き
標準エラー出力ファイル名	<input type="checkbox"/> 追加書き
終了判定	判定結果 <input type="text" value="しきい値による判定"/> 警告しきい値 <input type="text"/> 異常しきい値 <input type="text" value="0"/>
異常終了時リトライ	<input checked="" type="radio"/> しない <input type="radio"/> する
終了コード	<input type="text"/> 以上 <input type="text"/> 以下
最大リトライ回数	<input type="text" value="1"/> 回
リトライ間隔	<input type="text" value="1"/> 分
実行時のユーザー	
<input type="button" value="OK"/> <input type="button" value="キャンセル"/> <input type="button" value="ヘルプ"/>	

図 3-3 「詳細定義- [UNIX Job]」ダイアログボックスの「定義」タブの指定例（ジョブ定義スクリプトファイルパスを指定する場合）

ユニット名	sample
コメント	
実行エージェント	
定義 転送ファイル 属性	
コマンド文	
スクリプトファイル名	/home/user1/scripts/sct02.ash
パラメーター	param1 param2
環境変数	AJS_BJEX_STOP=TERM
環境変数ファイル名	
ワークパス	
実行優先順位	なし
標準入力ファイル名	
標準出力ファイル名	<input type="checkbox"/> 追加書き
標準エラー出力ファイル名	<input type="checkbox"/> 追加書き
終了判定	判定結果 <input type="text" value="しきい値による判定"/> 警告しきい値 <input type="text"/> 異常しきい値 <input type="text" value="0"/>
異常終了時リトライ	<input checked="" type="radio"/> しない <input type="radio"/> する
終了コード	<input type="text"/> 以上 <input type="text"/> 以下
最大リトライ回数	<input type="text" value="1"/> 回
リトライ間隔	<input type="text" value="1"/> 分
実行時のユーザー	
<input type="button" value="OK"/> <input type="button" value="キャンセル"/> <input type="button" value="ヘルプ"/>	

4

バッチジョブの実行と実行結果の確認

ここでは、バッチジョブの実行方法と、バッチジョブの実行結果を確認する方法について説明します。

4.1 バッチジョブの実行

ここでは、JP1/Advanced Shell で作成したバッチジョブを JP1/AJS から実行するときの流れと、実行の手順について説明します。

4.1.1 バッチジョブの実行の流れ

JP1/AJS を使用してジョブを実行する場合の、運用者の作業の流れを次に示します。

(1) ジョブの定義

JP1/AJS を使用してジョブを実行する場合には、「[3.1.1 ジョブネットを定義して実行する](#)」に示す手順に従ってジョブを定義しておきます。

(2) ジョブの実行

JP1/AJS を使用してジョブを実行する方法には、計画実行、確定実行および即時実行があり、おのおの方法に従って実行します。実行方法については、マニュアル「JP1/Automatic Job Management System 3 操作ガイド」を参照してください。

JP1/AJS を使用しない場合、コマンドプロンプトまたはシェルからコマンドを入力してジョブ（ジョブ定義スクリプト）を実行することもできます。

(3) ジョブネットの監視

JP1/AJS でジョブネットモニタを起動し、ジョブの実行状態を確認します。

(4) ジョブの再実行

ジョブの再実行が必要な場合、JP1/AJS - View の画面から再実行します。

4.1.2 バッチジョブを実行する

実行環境から JP1/AJS を使用して JP1/Advanced Shell のバッチジョブ業務を起動する方法について説明します。

JP1/AJS でのバッチジョブ業務の自動化の詳細については、JP1/AJS のマニュアルを参照してください。JP1/Advanced Shell のジョブをジョブネットに定義して実行する方法については、「[3.1.1 ジョブネットを定義して実行する](#)」を参照してください。

バッチジョブ業務を自動化することで、コストを削減できるだけでなく、少ない人員で確実にシステムを運用できます。JP1/AJS は、このような定型的なバッチジョブ業務を自動化するための製品です。JP1/AJS

は、複雑なバッチジョブ業務の組み合わせの自動化にも対応できます。また、JP1/AJS の運用時に JP1/Advanced Shell を使用することで、次のメリットが得られます。

- 一時ファイル機能によって、一時的に使用するファイルを割り当てて、ジョブ終了時またはジョブステップ終了時に削除できます。
- 外部スクリプトの呼び出しによって、ジョブの定義を業務間で共用できます。
- ジョブ定義スクリプトに対する変更、追加、または削除によって、柔軟なジョブ定義ができます。

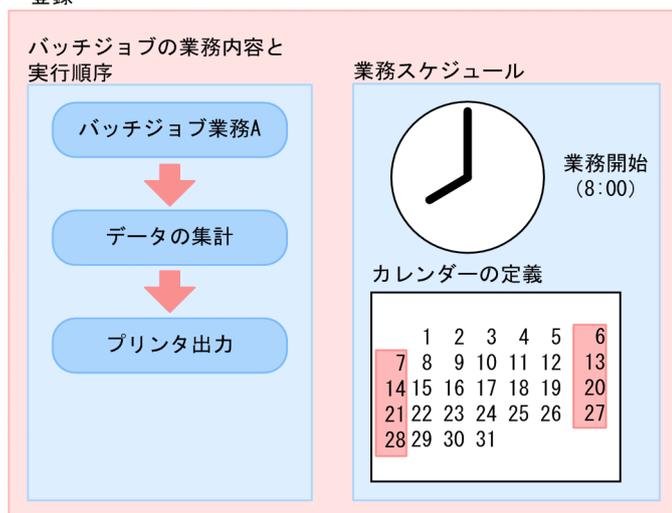
JP1/AJS を使用して、バッチジョブ業務を自動的に実行する場合、次に示す内容を定義する必要があります。

- バッチジョブ業務内容と順序
- バッチジョブ業務を実行するスケジュールまたはバッチジョブ業務の契機となる事象の登録

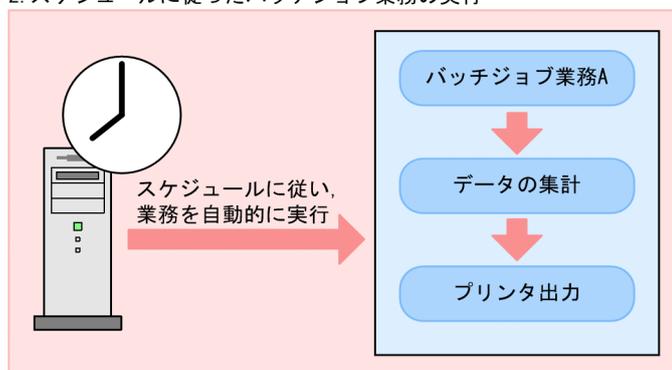
JP1/AJS を使用したバッチジョブ業務の自動化の概要を次の図に示します。図中の番号は、そのあとに示す説明の項番と対応しています。

図 4-1 JP1/AJS を使用したバッチジョブ業務の自動化の概要

1. バッチジョブの業務内容と実行順序、および業務スケジュールを登録



2. スケジュールに従ったバッチジョブ業務の実行



1. バッチジョブの業務内容と実行順序、および業務スケジュールを登録します。

4. バッチジョブの実行と実行結果の確認

2. 登録されたスケジュールに従って、バッチジョブ業務が自動的に実行されます。

(1) バッチジョブ業務と実行順序の定義

多くの業務は、決まった時間に定められた順序に従って実行されます。

例えば、売上傳票の集計は、次の順序で実行されます。

1. データベースからのデータの抽出
2. データのソート
3. プリント出力

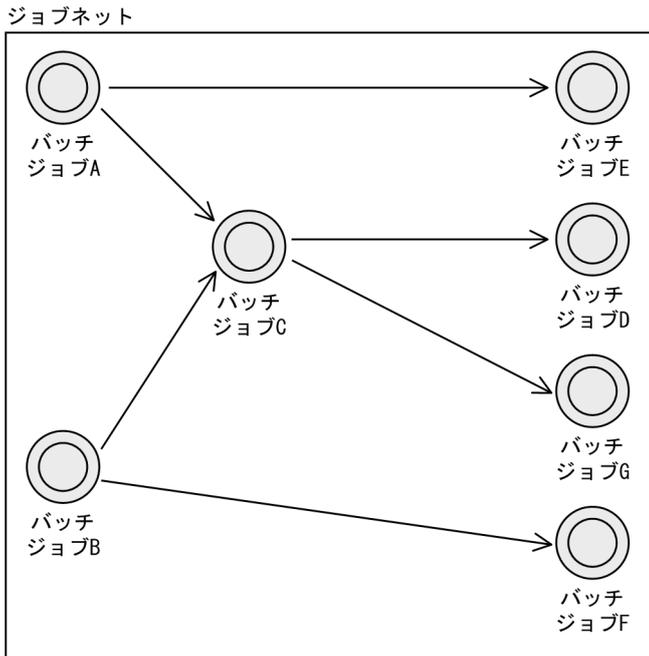
ジョブコントローラのジョブステップとして、1.~3.の流れをジョブ定義スクリプトファイルに定義することで自動化を実現できますが、12:00 にデータベースからデータを抽出するという作業は自動化できません。JP1/Advanced Shell と JP1/AJS でバッチジョブ業務と実行順序の定義を実行するには、業務を構成する一連の流れをジョブコントローラで定義し、それぞれのバッチジョブ業務と実行順序の定義の関係を JP1/AJS の実行順序、または実行時間として定義します。

JP1/AJS だけでもコマンド、アプリケーションプログラム、またはジョブ定義スクリプトなどのそれぞれの作業単位に分解すれば、JP1/Advanced Shell 相当のジョブを実現できます。これらを JP1/AJS でもジョブと呼びます。

JP1/Advanced Shell と JP1/AJS でバッチジョブ業務と実行順序を定義する場合には、JP1/AJS ではバッチジョブの実行順序をジョブネットで定義します。

JP1/Advanced Shell と JP1/AJS でバッチジョブ業務と実行順序の定義を行う場合のジョブネットを次の図に示します。

図 4-2 JP1/Advanced Shell と JP1/AJS でバッチジョブ業務と実行順序の定義を行う場合のジョブネット



(説明)

JP1/AJS のジョブネットで定義するバッチジョブの実行順序の流れを次に説明します。

- バッチジョブ A が終了した場合、バッチジョブ E を実行します。
- バッチジョブ A と B が終了した場合、バッチジョブ C を実行します。
- バッチジョブ C が終了した場合、バッチジョブ D と G を実行します。
- バッチジョブ B が終了した場合、バッチジョブ F を実行します。

(2) バッチジョブ業務と実行順序の定義スケジュールの定義

複数のバッチジョブ業務と実行順序の定義スケジュールの定義を自動化するには、この定義をいつ実行するかを決めるスケジュールの定義が必要です。

JP1/AJS のスケジュールの定義では、会社の営業日・休業日を設定したカレンダー、実行を開始する日時や実行間隔などを定義します。この定義に基づいて、JP1/AJS が実行予定を決め、その日時になると自動的に JP1/Advanced Shell のジョブ実行を始めます。

(3) バッチジョブ業務を開始する契機を登録する

ファイルの作成またはイベントの発生などをバッチジョブ業務開始の契機として登録できます。登録の結果、決まった時刻にバッチジョブ業務を開始するだけでなく、ファイルの作成またはイベントの発生など何らかの事象が起こった場合にも、バッチジョブ業務が開始できます。

4.2 バッチジョブの実行結果の確認

バッチジョブの実行結果は、ジョブ実行ログで確認できます。ここでは、ジョブ実行ログとは何か、ジョブ実行ログの確認、および削除の手順について説明します。

4.2.1 ジョブ実行ログ

ジョブ実行ログとは、バッチジョブの実行結果を通知する利用者向けのログ情報のことです。このログ情報はスプールジョブディレクトリ下のファイルに出力され、ユーザープログラムの標準出力以外はジョブ終了時に標準エラー出力に出力されます。標準エラー出力の結果は JP1/AJS - View などによって確認できます。

ジョブ実行ログには、次の情報が出力されます。

- バッチジョブの開始・終了メッセージ
- ジョブステップの開始・終了メッセージ
- ジョブ定義スクリプトの内容
- 実行したコマンドの結果
- 準備したファイルの状況、後処理の結果
- ユーザープログラムの標準出力 (stdout) ※1
- ユーザープログラムの標準エラー出力 (stderr) ※2
- カバレッジ取得に関するメッセージ

注※1

次のどちらかが指定されている場合に、ジョブ実行中に起動時の標準出力へ出力されます。

- adshexec コマンドの `-s` オプションまたは環境ファイルの `OUTPUT_STDOUT` パラメーターに `PARENT` を指定した場合
- ルートジョブが最小出力モードの場合

注※2

ルートジョブが最小出力モードの場合は、スプールジョブディレクトリ下のファイルには出力されませんが、ジョブ実行中に起動時の標準エラー出力へ出力されます。

なお、情報メッセージの一部を JOBLOG ファイルへ出力させないようにするには、`JOBLOG_SUPPRESS_MSG` パラメーターで設定します。指定できるメッセージなどについては、マニュアル「JP1/Advanced Shell」の「JOBLOG_SUPPRESS_MSG パラメーター」の説明を参照してください。

4.2.2 バッチジョブの実行結果を確認する

環境ファイルに設定されたスプールルートディレクトリに、ジョブごとのディレクトリを作成し、ジョブの実行結果を出力します。ジョブごとのディレクトリには、ジョブ実行ログやジョブステップのプログラムが出力したファイルが出力されます。

スプールディレクトリの構造を次に示します。なお、次の説明では、このマニュアルで説明するディレクトリおよびファイルだけを載せています。スプールディレクトリの構造の全体については、マニュアル「JP1/Advanced Shell」の「ジョブの実行結果をスプールに出力する」の説明を参照してください。

```
スプールルートディレクトリ
├─ロックファイル
└─スプールジョブディレクトリ
   ├─JOBLOG※1
   ├─JOBLOG_ジョブ識別子_通番
   ├─JOBLOG_子孫ジョブ起動順序通し番号※2
   ├─SCRIPT※1
   ├─SCRIPT_子孫ジョブ起動順序通し番号※2
   ├─STDERR※1
   ├─STDOUT※1
   ├─ステップ番号_ステップ名_STDOUT※1
   └─ステップ番号_ステップ名_STDERR※1
```

注※1

このファイルの内容はジョブ実行ログにも出力されます。ジョブ実行ログの出力内容については、「4.2.1 ジョブ実行ログ」を参照してください。

注※2

ジョブの実行中に一時的に作成されるファイルです。内容を次に示します。

ファイル名	説明
JOBLOG_子孫ジョブ起動順序通し番号	環境ファイルの SPOOLJOB_CHILDJOB パラメーターに MERGE を指定（子孫ジョブのスプールジョブをルートジョブのスプールジョブにマージする）した場合に出力される、マージ用の子孫ジョブのジョブ実行ログ
SCRIPT_子孫ジョブ起動順序通し番号	

UNIX の SIGKILL や Windows の TerminateProcess などによってジョブが即時終了した場合、これらのファイルがスプールジョブディレクトリ内に残ることがあります。スプールジョブディレクトリが不要になって削除する際は、これらのファイルも一緒に削除してください。

一時ファイル以外のファイルおよびディレクトリについて次に説明します。

(1) スプールルートディレクトリ

ディレクトリ名は環境ファイルの SPOOL_DIR パラメーターで設定します。

(2) スプールジョブディレクトリ

ジョブ通し番号と同じ名称のディレクトリで、ジョブ単位に作成されます。ジョブ終了時に「**ジョブ識別子-ジョブ名**」に変更されます。

蓄積されたスプールジョブは adshhk コマンドで削除できます。adshhk コマンドについては「[4.2.3 スプールジョブを削除する](#)」を参照してください。

ジョブ終了後のスプールジョブディレクトリのリネーム時、同一名のディレクトリがすでに存在すると、リネームに失敗します。リネームに失敗すると「ジョブ識別子」のディレクトリ名のまま残ります。ジョブの動作は完了しており後続ジョブが続行できるため rc=0 で終了しますが、ディレクトリが残ったままである間、そのジョブ識別子は使用できなくなると同時に adshhk コマンドによる削除がされなくなります。

(3) JOBLOG

コマンドの実行結果やファイルの割り当て結果など、ジョブの動作状況を示すメッセージが出力されます。

(4) JOBLOG_ジョブ識別子_通番

子孫ジョブのジョブ実行ログです。

子孫ジョブ起動時に次のどちらかの方法で、子孫ジョブを最小出力モードで指定した場合だけ作成されます。

- adshexec コマンドの -m オプションで MINIMUM を指定する
- adshscriptool コマンドの -m オプションで MINIMUM を指定する
- OUTPUT_MODE_CHILD パラメーターで MINIMUM を指定する

ただし、SPOOLJOB_CHILDJOB パラメーターに MERGE（子孫ジョブのスプールジョブをルートジョブのスプールジョブにマージする）を指定した場合は作成されません。

(5) SCRIPT

スクリプトイメージファイルです。最初に起動したジョブ定義スクリプトファイルと、#-adsh_script コマンドで指定した外部のジョブ定義スクリプトファイルの内容が出力されます。その他の、(ドット) コマンドなどで指定する外部のジョブ定義スクリプトファイルは出力されません。ログとしてジョブ定義スクリプトの内容を出力したい場合は、#-adsh_script コマンドを使用します。

なお、SPOOLJOB_CHILDJOB パラメーターで MERGE を指定し、ルートジョブを拡張出力モード、子孫ジョブを最小出力モードで動作する場合は、子孫ジョブの SCRIPT はルートジョブの SCRIPT にマージされません。詳細については、マニュアル「JP1/Advanced Shell」の「子孫ジョブのスプールジョブをルートジョブのスプールジョブへマージした場合」の説明を参照してください。

(6) STDERR

ジョブの標準エラー出力です。ルートジョブ起動時に次のどちらかの方法で、ルートジョブを最小出力モードで指定した場合は作成されません。

- adshexec コマンドの -m オプションで MINIMUM を指定する
- OUTPUT_MODE_ROOT パラメーターに MINIMUM を指定する

ファイルの先頭行には次に示すヘッダが出力されます。

```
***** 実行ジョブのSTDERRファイルの内容 *****
```

(7) STDOUT

ジョブの標準出力です。adshexec コマンドの -s オプションおよび環境ファイルの OUTPUT_STDOUT パラメーターに SPOOL を指定した場合に作成されます。ルートジョブ起動時に次のどちらかの方法で、ルートジョブを最小出力モードで指定した場合は作成されません。

- adshexec コマンドの -m オプションで MINIMUM を指定する
- OUTPUT_MODE_ROOT パラメーターに MINIMUM を指定する

ファイルの先頭行には次に示すヘッダが出力されます。

```
***** 実行ジョブのSTDOUTファイルの内容 *****
```

(8) ステップ番号_ステップ名_STDOUT

ジョブステップを定義した場合の、該当するジョブステップ中の標準出力です。ジョブステップ名が 8 バイトを超える場合、ファイル名に含むステップ名は、ジョブステップ名の最初の 8 バイトになります。

adshexec コマンドの -s オプションおよび環境ファイルの OUTPUT_STDOUT パラメーターに SPOOL を指定した場合に作成されます。ルートジョブ起動時に次のどちらかの方法で、ルートジョブを最小出力モードで指定した場合は作成されません。

- adshexec コマンドの -m オプションで MINIMUM を指定する
- OUTPUT_MODE_ROOT パラメーターに MINIMUM を指定する

(9) ステップ番号_ステップ名_STDERR

ジョブステップを定義した場合の、該当するジョブステップ中の標準エラー出力です。ジョブステップ名が 8 バイトを超える場合、ファイル名に含むステップ名は、ジョブステップ名の最初の 8 バイトになります。

ルートジョブ起動時に次のどちらかの方法で、ルートジョブを最小出力モードで指定した場合は作成されません。

- adshexec コマンドの -m オプションで MINIMUM を指定する
- OUTPUT_MODE_ROOT パラメーターに MINIMUM を指定する

4.2.3 スプールジョブを削除する

スプールに格納されたスプールジョブは、スプールディレクトリに保存されたまま、増加します。このため、定期的に古いスプールジョブを削除してディスクの空き容量を増やす必要があります。

- スプールジョブの削除方法

スプールジョブを削除するためには、次の adshhk コマンドを入力します。adshhk コマンドの指定方法については、マニュアル「JP1/Advanced Shell」の「adshhk コマンド」の説明を参照してください。

```
adshhk 対象リストファイル名 レポートファイル名 ログファイル名 [日数]
```

対象リストファイル名のファイルには、削除したいスプールジョブのあるスプールディレクトリ名などを adshhk コマンドの実行前に記載する必要があります。

レポートファイル名のファイルに adshhk コマンドの実行結果が出力されます。実行結果は、トレースログにも出力されます。

ログファイル名のファイルには、エラーメッセージを出力します。

adshhk コマンド実行日の前日を基点にして、指定した日数以前のスプールジョブを削除します。例えば、2 を指定した場合、昨日（基点）とおとといの2日間の指定となるため、おととい以前のスプールジョブを削除します。なお、当日のスプールジョブは削除できません。

- adshhk コマンドで作成されるレポートファイル

adshhk コマンドを実行した場合、実行結果がレポートファイルに出力されます。次のレポートファイルの例では、先頭行にヘッダ情報が出力されています。

```
"jobid","jobname","rc","start date","end date","act","info","spool","target
days","execute date"
"000056","JOB001","1","2011/06/13 09:03:31","2011/06/13 09:03:31","delete","","C:
¥Documents and Settings¥All Users¥Documents¥Hitachi¥jplase¥jplase¥spool","15","2011/06/30
18:19:58"
:
```

(凡例)

実行結果の1行目の見出しの意味は次のとおりです。2行目以降には対応した項目の値が表示されます。

見出し	意味
jobid	ジョブ識別子
jobname	ジョブ名

見出し	意味
rc	ジョブの終了コード
start date	ジョブの実行開始日時 (yyyy/mm/dd hh:mm:ss の形式)。 デバッグモードで起動したときにデバッガ自身に割り当てるスプールは、ジョブの実行結果ではなくデバッガのログ出力用のため、デバッグの開始日時が出力されます。
end date	ジョブの終了日時 (yyyy/mm/dd hh:mm:ss の形式)。 デバッグモードで起動したときにデバッガ自身に割り当てるスプールは、ジョブの実行結果ではなくデバッガのログ出力用のため、ジョブ終了日時は出力されません。
act	適用した動作 (keep : 保存, delete : 削除, error : 削除処理中にエラー発生)
info	エラーの詳細情報
spool	スプールディレクトリ
target days	対象日数
execute date	コマンドの実行開始日時 (yyyy/mm/dd hh:mm:ss の形式)

注意事項

スプールのディレクトリの下に、JP1/Advanced Shell が作成するファイルやディレクトリではないユーザー独自のファイルやディレクトリが存在する場合、adshhk コマンドは KNAX4419-E メッセージなどを出力して終了します。

4.3 バッチジョブの強制終了

ジョブの強制終了には次の 2 とおりの方法があります。

- JP1/AJS からジョブを起動している場合、JP1/AJS の強制終了操作を実行します。
JP1/AJS から Windows または UNIX で実行されたジョブアイコンのジョブを強制終了する場合、環境変数 `AJS_BJEX_STOP=TERM` を設定しておく必要があります。Windows または UNIX で実行されたジョブアイコンのジョブの詳細については、「[3.1.1 ジョブネットを定義して実行する](#)」を参照してください。
- `adshexec` コマンドのプロセスに対して、終了要求シグナルを送付します。Windows の場合は、`taskkill` コマンドなどを用いて `adshexec` のプロセスを終了させます。

ジョブを強制終了すると、ジョブコントローラは実行中の子プロセスまたは子孫プロセスを強制終了します。詳細は、「[4.3.1 子プロセスまたは子孫プロセスの強制終了](#)」を参照してください。

子プロセスまたは子孫プロセスを強制終了したあと、割り当てたファイルの後処理をして後続のジョブステップ・コマンドを一切実行しないで終了します。後続ジョブステップの `run` 属性に `abnormal` や `always` が指定されていても実行しません。ジョブを強制終了すると、UNIX では `adshexec` コマンドがシグナルによってエラー終了します。UNIX での `SIGTERM` 受信時のジョブの動作については、マニュアル「JP1/Advanced Shell」の「シグナル受信時の動作」の説明を参照してください。Windows での強制終了時のジョブの動作については、マニュアル「JP1/Advanced Shell」の「強制終了時のジョブの動作」の説明を参照してください。

注意事項

Windows の場合、`adshexec` コマンドの起動時に `adshexecsub` コマンドをあわせて起動し、`adshexec` コマンドを強制終了すれば `adshexecsub` コマンドも終了します。そのため、`adshexecsub` コマンドは強制終了しないでください。`adshexecsub` コマンドを強制終了すると、次の現象が発生する場合があります。

- 実行中の子孫プロセスが終了されません。
- 一時ファイルが残ったままとなる場合があります。

これらの現象が発生した場合は、`taskkill` コマンドやタスクマネージャーを使用して子孫プロセスを強制終了させ、一時ファイルを手動で削除してください。

注意事項

Windows 環境の JP1/Advanced Shell では、孫プロセスの強制終了のためにジョブオブジェクトを使用しているため、次の 2 点に注意してください。

- JP1/Advanced Shell から生成した子プロセスをジョブオブジェクトに関連づけることはできません。
- JP1/Advanced Shell のプロセスが、すでにジョブオブジェクトに関連づけられていた場合、ジョブの強制終了で JP1/Advanced Shell の子プロセスが生成したプロセスは終了しません。

注意事項

Windows では、子孫プロセスを生成する外部コマンドを実行するジョブを強制終了した場合、孫以下のプロセスが同時に 256 個以上存在すると、メッセージ KNAX6381-E を出力してスプールジョブディレクトリの名称変更が失敗することがあるため、次の 3 点に注意してください。

- 失敗したスプールジョブディレクトリを参照する場合は、直後のメッセージ KNAX6382-I に出力されるディレクトリ名を参照してください。
- 名称変更失敗したスプールジョブディレクトリは `adshhk` コマンドでは削除されません。削除する場合は、手動で削除してください。
- 実行環境でスプールジョブディレクトリの名称変更失敗したジョブは、`adshevtout` コマンドでジョブ定義スクリプト稼働実績情報を出力しません。

4.3.1 子プロセスまたは子孫プロセスの強制終了

ジョブを強制終了すると、ジョブコントローラは子プロセスまたは子孫プロセスを強制終了してからジョブを終了します。

4.3.2 Ctrl+C などの操作に関する注意事項 (Linux の場合)

非端末入力モードでジョブを実行した場合、ルートジョブ、子孫ジョブ、およびほかに起動した外部コマンドを「Ctrl+C」や「Ctrl+¥」などの操作で一度に強制終了させることができないときがあります※。これらのジョブおよびコマンドを一度にすべて強制終了させたいときは、ログインシェル直下のルートジョブに対して、`kill` コマンドで SIGTERM などの終了要求シグナルを送信してください。

注※

非端末入力モードでジョブを実行した場合、`adshexec` コマンドのプロセスとその子プロセスとは、別々のプロセスグループになります。そのため、ログインシェルからジョブを実行中に「Ctrl+C」や「Ctrl+¥」などの操作を実行した場合、現在フォアグラウンドにいるプロセスグループに対してだけ、SIGINT や SIGQUIT が送られます。

シグナルを受信したジョブの子孫プロセスとして動作するジョブおよび外部コマンドは強制終了されますが、親プロセスやそれより上位のプロセスとして動作するジョブおよび外部コマンドは強制終了されません。

付録

付録 A もっと使いこなすには？

JP1/Advanced Shell をもっと使いこなすには、マニュアル「JP1/Advanced Shell」を参照してください。

付録 B このマニュアルの参考情報

このマニュアルを読むに当たっての参考情報を示します。

付録 B.1 関連マニュアル

関連マニュアルを次に示します。必要に応じてお読みください。

JP1/Advanced Shell 関連

- JP1 Version 11 JP1/Advanced Shell (3021-3-B32)

JP1/AJS 関連

- JP1 Version 11 ジョブ管理 基本ガイド (ジョブスケジューラー編) (3021-3-B11)
- JP1 Version 11 JP1/Automatic Job Management System 3 設計ガイド (システム構築編) (3021-3-B13)
- JP1 Version 11 JP1/Automatic Job Management System 3 構築ガイド (3021-3-B15)
- JP1 Version 11 JP1/Automatic Job Management System 3 トラブルシューティング (3021-3-B17)
- JP1 Version 11 JP1/Automatic Job Management System 3 操作ガイド (3021-3-B18)

付録 B.2 マイクロソフト製品の表記

このマニュアルでは、マイクロソフト製品の名称を次のように表記しています。

このマニュアルでの表記		正式名称
Windows Server*	Windows Server 2012	Microsoft(R) Windows Server(R) 2012 Standard
		Microsoft(R) Windows Server(R) 2012 Datacenter
		Microsoft(R) Windows Server(R) 2012 R2 Standard
		Microsoft(R) Windows Server(R) 2012 R2 Datacenter
	Windows Server 2008	Microsoft(R) Windows Server(R) 2008 R2 Datacenter
		Microsoft(R) Windows Server(R) 2008 R2 Enterprise
Microsoft(R) Windows Server(R) 2008 R2 Standard		
Windows*	Windows 10	Windows(R) 10 Home(32 ビット版)
		Windows(R) 10 Pro (32 ビット版)
		Windows(R) 10 Enterprise (32 ビット版)

このマニュアルでの表記		正式名称
Windows*	Windows 10	Windows(R) 10 Home(64 ビット版)
		Windows(R) 10 Pro(64 ビット版)
		Windows(R) 10 Enterprise(64 ビット版)
	Windows 8	Windows(R) 8.1(32 ビット版)
		Windows(R) 8.1 Pro(32 ビット版)
		Windows(R) 8.1 Enterprise(32 ビット版)
		Windows(R) 8.1(64 ビット版)
		Windows(R) 8.1 Pro(64 ビット版)
		Windows(R) 8.1 Enterprise(64 ビット版)
		Windows(R) 8(32 ビット版)
		Windows(R) 8 Pro(32 ビット版)
		Windows(R) 8 Enterprise(32 ビット版)
		Windows(R) 8(64 ビット版)
		Windows(R) 8 Pro(64 ビット版)
		Windows(R) 8 Enterprise(64 ビット版)
	Windows 7	Microsoft(R) Windows(R) 7 Enterprise
		Microsoft(R) Windows(R) 7 Professional
		Microsoft(R) Windows(R) 7 Ultimate

注※ Windows Server および Windows を総称して Windows と表記することがあります。

付録 B.3 このマニュアルで使用している記号

このマニュアルの文法説明で使用する記号を次に示します。

記号	意味
[]	GUI 説明の場合、メニュー項目、ダイアログボックス、ボタンなどを示します。 例 [ファイル] - [新規作成] を選択する。 上記の例では、メニューバーの [ファイル] を選択して、ドロップダウンリストの [新規作成] を選択することを示します。
< >	各項目を記述するときに従わなくてはならない構文要素記号*を示します。
+	この記号の直前に示された項目を繰り返して複数指定ができます。または、この記号の前後の項目が同時に設定されていることを示します。

記号	意味
+	例 {A B} + 「A または B を任意の順序で 1 つ以上指定する」ことを示します。 [CR] + [LF] [CR] と [LF] が同時に設定されていることを示します。
~	この記号の直前に示されている項目を、この記号に続く < >, 《 》, (())などの文法規則に従って記述する必要があることを示します。
(())	指定できる値の範囲を示します。
…および… (点線)	この記号の直前に示された項目を繰り返して複数個指定できることを示します。 例 A, B, … 「A のあとに B を必要個数指定する」ことを示します。
太字	可変および強調を示します。

注※

このマニュアルの構文要素記号を次に示します。

構文要素記号	指定できる文字の内容
<符号なし整数>	<数字> +

付録 B.4 Windows の場合の「インストール先フォルダ」について

このマニュアルで使用している「**インストール先フォルダ**」は、特に断りがないかぎり JP1/Advanced Shell のインストール先フォルダを示しています。製品を初期設定のままインストールした場合のインストール先フォルダは次のとおりです。

x86 環境の場合

システムドライブ:%Program Files%Hitachi%JP1AS

x64 環境の場合

システムドライブ:%Program Files(x86)%Hitachi%JP1AS

付録 B.5 このマニュアルで使用する「共通アプリケーションフォルダ」について

このマニュアルで使用している「**共通アプリケーションフォルダ**」について次に示します。

Windows 10, Windows Server 2012, Windows 8, Windows 7, または Windows Server 2008 の場合の例

システムドライブ:¥ProgramData

付録 B.6 このマニュアルで使用する「共有ドキュメントフォルダ」について

このマニュアルで使用している「**共有ドキュメントフォルダ**」について次に示します。

Windows 10, Windows Server 2012, Windows 8, Windows 7, または Windows Server 2008 の場合の例

システムドライブ:¥Users¥Public¥Documents

付録 B.7 このマニュアルで使用する Windows のメニュー名について

このマニュアルで使用している Windows のメニュー名の表記は、次の OS を前提としています。

Windows 7, Windows Server 2008

Windows 10, Windows Server 2012 または Windows 8 の場合は [スタート] メニューが表示されないため、画面左下から表示できる [スタート] 画面からメニューを選択してください。

付録 B.8 フォルダおよびディレクトリの表記

Windows と UNIX で共用する部分では、ディレクトリという用語を使用しています。Windows 限定の記載がある場合には、フォルダを使用します。

上記に伴って、ディレクトリの区切りには、「/」を使用します。Windows 特有で使用する場合は、フォルダの区切りには、「¥」を使用します。

付録 B.9 製品名の表記

このマニュアルでは、製品名を次のように表記しています。

このマニュアルでの表記		正式名称
JP1/Advanced Shell		JP1/Advanced Shell
JP1/AJS	JP1/AJS3	JP1/Automatic Job Management System 3 - Agent JP1/Automatic Job Management System 3 - Manager JP1/Automatic Job Management System 3 - View

このマニュアルでの表記		正式名称	
JP1/AJS - Agent	JP1/AJS3 - Agent	JP1/Automatic Job Management System 3 - Agent	
JP1/AJS - Manager	JP1/AJS3 - Manager	JP1/Automatic Job Management System 3 - Manager	
JP1/AJS - View	JP1/AJS3 - View	JP1/Automatic Job Management System 3 - View	
UNIX	Linux	CentOS 6	CentOS 6
		CentOS 7	CentOS 7
		Oracle Linux 6	Oracle Linux(R) 6
		Oracle Linux 7	Oracle Linux(R) 7
		Red Hat Enterprise Linux 6 (x64)	Red Hat Enterprise Linux Server 6 (64-bit x86_64)
		Red Hat Enterprise Linux 7	Red Hat Enterprise Linux Server 7 (64-bit x86_64)
		SUSE Linux 12	SUSE Linux(R) Enterprise Server 12(x64)

付録 B.10 KB (キロバイト) などの単位表記

1KB (キロバイト), 1MB (メガバイト), 1GB (ギガバイト), 1TB (テラバイト) はそれぞれ $1,024$ バイト, $1,024^2$ バイト, $1,024^3$ バイト, $1,024^4$ バイトです。

付録 C 用語解説

このマニュアルで使用する用語について解説します。

(英字)

export パラメーター

環境ファイルに設定するパラメーターのうち、コマンド起動時に環境変数を設定するために指定するパラメーターです。

JP1/Advanced Shell

バッチジョブのためのジョブ定義スクリプトを作成・実行するための製品です。JP1/Advanced Shell は、JP1/Advanced Shell と JP1/Advanced Shell - Developer とに分けられます。JP1/Advanced Shell では、バッチジョブのためのジョブ定義スクリプトを実行でき、狭義には JP1/Advanced Shell を実行環境と呼びます。同じジョブ定義スクリプトのバッチジョブを Windows および UNIX の両方で実行できます。

JP1/Advanced Shell - Developer

バッチジョブのためのジョブ定義スクリプトを開発するための製品です。ジョブ定義スクリプトを開発するため、開発環境と呼ぶこともあります。

JP1/AJS3

JP1/Automatic Job Management System 3 の略で、JP1/AJS2 の後継製品です。JP1/Advanced Shell は、JP1/AJS3 と連携することで、複数の PC 間での分散処理が実現できます。

UNIX 互換コマンド

JP1/Advanced Shell では、UNIX でよく使用されるコマンドの一部を使用できます。Windows 環境でも使用でき、UNIX から Windows への移行性を向上できます。ls コマンドなどがあります。

(ア行)

一時ファイル

ジョブ実行時に一時的に使用するファイルです。ジョブまたはジョブステップによって作成され、ジョブ終了時には自動的に削除されます。#-adsh_file_temp コマンドで定義できます。

エディタ

開発環境に付属するさまざまな機能を利用して、効率良くジョブ定義スクリプトを作成できます。

オプション

コンピュータの入力装置から入力する指示に対して、選択的な機能を付け加えます。このことをオプションといいます。

JP1/Advanced Shell では、1 個のハイフン (-) に続く 1 文字のコマンド引数をショートオプション、連続する 2 個のハイフン (--) に続くコマンド引数をロングオプションと呼んでいます。

オプションの右側に指定する引数のことをオプションの値と呼びます。

オペランド

コマンドラインに指定するコマンド引数のうち、オプション名とオプション値のほかに指定する、規定のコマンド引数のことです。また、パラメーターの値もオペランドと呼びます。

(カ行)

開発環境

JP1/Advanced Shell - Developer が提供する、バッチ処理のためのジョブ定義スクリプトを開発するための環境です。

外部コマンド

シェルの組み込みコマンドではない、UNIX 互換コマンド、OS が提供するコマンドおよびユーザーによって作成される実行ファイルやプログラムのことを指します。

カスタムジョブ

ある特定の機能を持つジョブを JP1/AJS で実行できるように定義したジョブです。JP1/Advanced Shell で JP1/AJS のカスタムジョブ機能を利用するには、JP1/Advanced Shell 用のカスタムジョブコンポーネントが必要です。

カバレッジ情報

プログラムのテストがどれだけ網羅されているかを示す指標です。C0（ステートメントカバレッジ情報）と C1（ブランチカバレッジ情報）とがあります。

C0 は、ジョブ定義スクリプトのコマンドをどれだけ実行したかの指標（%）です。

C1 は、ジョブ定義スクリプトの分岐をどれだけ実行したかの指標（%）です。

環境情報

JP1/Advanced Shell を起動する前に設定が必要な、環境変数や環境ファイルのパラメーターなどの情報のことです。

環境設定パラメーター

環境ファイルに設定するパラメーターのうち、「#-adsh_conf パラメーター 値」という形式で指定し、JP1/Advanced Shell の実行環境を定義するパラメーターのことです。

環境ファイル

環境情報を格納したファイルのことです。

環境変数

ユーザーが設定できるシステムの各種の設定を格納した変数のことです。

組み込みコマンド

シェル本体に組み込まれたコマンドであり、シェル自身によって実行できます。JP1/Advanced Shell ではシェル標準コマンド（特殊組み込みコマンドおよび正規組み込みコマンド）、シェル拡張コマンドを提供します。特殊組み込みコマンドは、コマンドの構文を誤るとコマンドを実行しているシェルが終了する特徴を持ちます。正規組み込みコマンドは、コマンドの構文を誤ってもコマンドを実行しているシェルは終了しないで継続します。

コマンド

シェル、コマンドプロンプトまたはジョブ定義スクリプトから実行する JP1/Advanced Shell で使用できるコマンドの総称のことです。

コマンドプロンプト

Windows 環境でコマンドの入力を促すものです。

コマンドライン

ユーザーがコマンドを入力するための行です。Windows では、コマンドプロンプトにあり、行は>の次から入力します。UNIX では、シェルにあり、行は%の次から入力します。

(サ行)

シェル

コンピュータの入力装置から入力された指示を解釈して、OS に伝えるプログラムのことです。

シェルスクリプト

テキストファイルにコマンドを並べて記載しておき、シェルからそのコマンドを続けて実行できるようにしたテキストファイルを、シェルスクリプトといいます。JP1/Advanced Shell のシェルスクリプトは、Windows 環境と UNIX 環境で実行でき、ジョブ定義スクリプトといいます。

シェル標準コマンド

シェル本体に組み込まれたコマンドであり、シェル自身のプロセスで実行されます。ジョブ定義スクリプトで使用できるコマンドです。

シェル変数

ジョブ定義スクリプト内で値を代入する領域のことです。変数の作成変数の値を参照できます。

シグナル

UNIX の場合にプロセス間で非同期イベントの発生を伝える機構です。JP1/Advanced Shell では、ジョブの強制終了などに使用します。

システム実行ログ

システム管理者が JP1/Advanced Shell によるジョブ実行状況を統合管理するため、ジョブコントローラから出力されるログのことです。複数のジョブコントローラが出力するログを、1つのログにまとめて出力できます。

子孫ジョブ

ルートジョブの子孫プロセスとして実行されるジョブ定義スクリプトのうち、次のどれかのパラメーターの指定、またはパラメーターのデフォルト定義によって実行されたジョブのことです。

- CHILDJOB_EXT パラメーター
- CHILDJOB_PGM パラメーター
- CHILDJOB_SHEBANG パラメーター

実行環境

JP1/Advanced Shell が提供する、バッチ業務を実行するための環境です。狭義には、JP1/Advanced Shell のことです。

終了コード

ジョブ定義スクリプトまたはコマンドを実行した場合に返信されるコードのことです。

条件式

ジョブ定義スクリプトで使用する、数値比較、文字列比較、ファイル属性、論理結合の演算子および三項演算子を使って表す計算式のことです。

ジョブコントローラ

ジョブ実行時にジョブをコントロールするためのプログラムです。adshexec コマンドがジョブコントローラに該当します。

ジョブ識別子

ジョブ実行時に JP1/Advanced Shell が与える 000001 から 999999 の識別番号です。各ジョブには別々の識別子が与えられ、ジョブ識別子によって一意にジョブを特定できます。ジョブ識別子を 999999 まで使用すると、ラップアラウンドして 000001 以降の未使用のジョブ識別子を使用します。

ジョブ実行ログ

ジョブやジョブステップの開始・終了メッセージなどの、ジョブが出力したメッセージの集まりです。ジョブ実行ログの内容は、ジョブ終了時にジョブコントローラの標準エラー出力に出力します。

ジョブ情報

ジョブに付随した情報のことです。ジョブ名、ジョブ識別子およびジョブステップ名などがあります。

ジョブスケジューラ

ジョブのスケジュールを行う製品であり、JP1/Advanced Shell では関連製品として JP1/AJS と連携できます。

ジョブステップ

ある業務（仕事）を行うための最小単位で、JP1/Advanced Shell ではジョブ定義スクリプトに記載されたジョブ内で、ある処理の単位で区切った範囲をいいます。ジョブステップの集まりがジョブになります。#-adsh_step_start コマンド、#-adsh_step_error コマンド（省略できます）、および#-adsh_step_end コマンドを記述して定義できます。

ジョブ定義スクリプトファイル

ジョブ定義スクリプトで作成した、ジョブを定義したプログラムのファイルのことです。

ジョブネット

実行順序を関連づけたジョブの集まりです。ジョブネット内のジョブは、あらかじめ定義した実行順序に従って自動的に実行されます。ジョブネットは、JP1/AJS の機能です。

シンボリックリンク

実際のファイルパスを格納したファイルを使ってリンクすることです。

スクリプト

テキストファイルにコマンドを並べて記載しておき、シェルからそのコマンドを続けて実行できるようにしたテキストファイルを、スクリプトといいます。JP1/Advanced Shell のスクリプトは、Windows 環境と UNIX 環境で実行でき、ジョブ定義スクリプトともいいます。

スクリプト拡張コマンド

ジョブ定義スクリプトで実行するコマンドです。通常のシェルスクリプトのコマンドに対してバッチジョブの実行を制御するための機能を付け加えたコマンドです。ジョブ実行制御コマンドともいいます。JP1/Advanced Shell では、`#-adsh` で始まるコマンドがあります。

スクリプトファイル

作成したスクリプトを保存したファイルです。

スクリプト予約語コマンド

ジョブ定義スクリプトで予約語として使用できるコマンドのことです。time コマンドがあります。

スプール

JP1/Advanced Shell でジョブの実行結果やジョブ実行ログを格納する場所です。

スプールジョブ

スプールディレクトリに作成されたジョブごとの実行結果のことです。

正規組み込みコマンド

シェル標準コマンドの組み込みコマンドの一種です。コマンドの構文を誤ってもコマンドを実行しているシェルが終了しないコマンドです。

(タ行)

ダイアログボックス

ユーザーに応答を促すウィンドウのことです。

通常ファイル

ジョブ定義スクリプトの入力および出力に使用するファイルです。ジョブ終了後にジョブ結果として残すファイルですが、ジョブの実行中に削除することもできます。`#-adsh_file` コマンドまたは `adshfile` コマンドで定義できます。

定義ファイル

トラブルシューティングのための資料を採取するディレクトリを定義しておくファイルです。

デバッグ

開発環境で作成したジョブ定義スクリプトをテストして不具合を調査するプログラムです。Windows 環境では、JP1/Advanced Shell エディタのデバッグ機能を使います。UNIX 環境では、`adshexec` コマンドに `-d` オプションを指定してデバッグを起動します。

デバッグ

開発環境で作成したジョブ定義スクリプトをテストして不具合を調査することです。デバッグを起動して調査します。

特殊組み込みコマンド

シェル標準コマンドの組み込みコマンドの一種です。コマンドの構文を誤るとコマンドを実行しているシェルが終了するコマンドです。

トレースログ

JP1/Advanced Shell でトラブルが発生した場合に、問題点を解明するために採取する情報のことです。

(ハ行)

バッチ業務サーバ

JP1/Advanced Shell をインストールしてバッチジョブを実行するサーバのことです。JP1/AJS を使用する場合、JP1/AJS - Agent または JP1/AJS - Manager をインストールします。

バッチジョブ

バッチ処理で実行するジョブのことです。

バッチ処理

収集したデータやトランザクションを 1 日分、1 週間分、1 か月分などにまとめて一括処理することです。

引数

コマンドラインやジョブ定義スクリプトにコマンドを実行する記述をする場合、コマンド名の後ろに区切り文字で区切って指定する項目の総称を引数といいます。

標準エラー出力 (stderr)

プログラムがエラーなどのメッセージを出力するストリームです。

標準出力 (stdout)

プログラムがデータを出力するストリームです。

標準入力 (stdin)

プログラムへデータを入力するストリームです。

ファイルの割り当て

JP1/Advanced Shell では、ファイルの後処理を登録することを含めてファイルの割り当てといいます。

ブレークポイント

ジョブ定義スクリプトの開発時にジョブ定義スクリプトの動作状態を確認するために、ジョブ定義スクリプト中に挿入される強制実行停止コードのことです。ブレークポイントではデバッガが処理を停止するため、開発者は停止直前の変数を確認できます。

プログラム出力データファイル

ユーザープログラムの出力結果をシステム実行ログと同様に一元管理するために、JP1/Advanced Shell が自動的にファイル名を作成して、ユーザープログラムが実行結果を出力するためのファイルです。

変数

ジョブ定義スクリプト内で値を扱うために使用する領域および配列のことです。変数にはシェル変数および環境変数も含まれます。

(ラ行)

リダイレクト

ジョブ定義スクリプトでは、コマンド実行前に実行結果の出力先の変更やコマンド実行に必要な情報の入力先を変更できます。これをリダイレクトといいます。通常、標準入力にはキーボードに、標準出力は画面に割り当てられていますが、リダイレクトではこれらの割り当てを変更します。

ルートジョブ

JP1/AJS やログインシェルなどから実行するジョブのうち、子孫ジョブ以外のジョブのことです。

ログ

コンピュータが出力する記録情報のことです。ログには記録した時間やメッセージなどが出力されます。

索引

A

- ADSH_ENV [環境変数] 37
- ADSH_JOBRC_FATAL [環境変数] 37, 38
- AJS_BJEX_STOP [環境変数] 37

E

- export パラメーター [用語解説] 68

J

- JP1/Advanced Shell [用語解説] 68
- JP1/Advanced Shell で使用できるファイルを確認する 14
- JP1/Advanced Shell のインストール 18
- JP1/Advanced Shell をインストールする (Windows の場合) 18
- JP1/Advanced Shell - Developer [用語解説] 68
- JP1/Advanced Shell と JP1/AJS でバッチジョブ業務と実行順序の定義を行う場合のジョブネット 51
- JP1/Advanced Shell の目的 3
- JP1/AJS3 [用語解説] 68
- JP1/AJS の環境情報を設定する (JP1/AJS を使用する場合) 40
- JP1/AJS のジョブに関する運用者の作業 49
- JP1/AJS を使用したバッチジョブ業務の自動化の概要 49

K

- KB (キロバイト) などの単位表記について 67

P

- PC ジョブ 40

U

- UNIX 互換コマンド [用語解説] 68
- UNIX 互換コマンドを使用するための設定をする 25
- UNIX ジョブ 40

い

- 一時ファイル 23
- 一時ファイル [用語解説] 68
- インストール前の準備 13
- インストールに必要なメモリ, ディスク容量 13
- インストールの流れ (Linux の場合) 12
- インストールの流れ (Windows の場合) 12

え

- エディタ [用語解説] 68

お

- オプション [用語解説] 68
- オペランド [用語解説] 68

か

- 開発環境 [用語解説] 68
- 外部コマンド [用語解説] 68
- 拡張出力モード 31
- カスタムジョブ 40
- カスタムジョブ [用語解説] 68
- カバレッジ情報 [用語解説] 68
- 環境情報 [用語解説] 68
- 環境設定パラメーター [用語解説] 68
- 環境ファイル [用語解説] 68
- 環境ファイルを設定する 22
- 環境変数 [用語解説] 68
- 環境変数 ADSH_JOBRC_FATAL (ジョブ続行不可エラー発生時の終了コードを設定する) 38
- 環境変数を設定する 37
- 関連マニュアル 63

き

- 業務への応用例 5

<

- 組み込みコマンド [用語解説] 68

こ

- このマニュアルでの表記 66
- このマニュアルの参考情報 63
- コマンド [用語解説] 68
- コマンドプロンプト [用語解説] 68
- コマンドライン [用語解説] 68

さ

- 最小出力モード 31

し

- シェル [用語解説] 68
- シェルスクリプト 3
- シェルスクリプト [用語解説] 68
- シェル標準コマンド [用語解説] 68
- シェル変数 [用語解説] 68
- シェルを設定する 26
- シグナル [用語解説] 68
- システム環境ファイル 22
- システム実行ログ 23
- システム実行ログ [用語解説] 68
- 子孫ジョブ [用語解説] 68
- 子孫ジョブとして起動するファイルを定義する 27
- 実行環境 [用語解説] 68
- 実行環境から JP1/AJS を使用してジョブを起動する 49
- 終了コード [用語解説] 68
- 条件式 [用語解説] 68
- ジョブ環境ファイル 22
- ジョブ強制終了時にユーザー固有の後処理を実行する 35
- ジョブコントローラ [用語解説] 68
- ジョブ識別子 [用語解説] 68
- ジョブ実行ログ 53
- ジョブ実行ログ [用語解説] 68
- ジョブ情報 [用語解説] 68
- ジョブスケジューラ [用語解説] 68
- ジョブステップ [用語解説] 68

ジョブ続行不可エラー発生時の終了コードを設定する 38

- ジョブ定義スクリプト 3
- ジョブ定義スクリプトファイル [用語解説] 68
- ジョブネット 51
- ジョブネット [用語解説] 68
- ジョブネットの監視 49
- ジョブネットを定義して実行する 40
- ジョブの再実行 49
- ジョブの実行 49
- ジョブの実行結果をスプールに出力する 54
- ジョブの種類ごとのジョブ実行ログの出力内容 28
- ジョブの定義 49
- ジョブを続行できないエラーが発生したときの終了コードを定義する 37
- シンボリックリンク [用語解説] 68

す

- スクリプト [用語解説] 68
- スクリプト開発部品 [用語解説] 68
- スクリプト拡張コマンド [用語解説] 68
- スクリプト拡張コマンドの終了コードを定義する 35
- スクリプト形式の UNIX 互換コマンドを使うための準備 (Windows の場合) 26
- スクリプトファイル [用語解説] 68
- スクリプト予約語コマンド [用語解説] 68
- スプール 23
- スプール [用語解説] 68
- スプールジョブ [用語解説] 68
- スプールジョブ作成抑止機能の使用有無を決定する 31
- スプールジョブを削除する 57
- スプールの出力情報を定義する 31

せ

- 正規組み込みコマンド [用語解説] 68
- 前提 OS 13
- 前提製品のインストール 17
- 前提となる OS の言語設定 13

た

ダイアログボックス〔用語解説〕 68

つ

通常ファイル〔用語解説〕 68

て

定義ファイル〔用語解説〕 68

ディレクトリの表記 66

デバッグ〔用語解説〕 68

デバッグ〔用語解説〕 68

と

特殊組み込みコマンド〔用語解説〕 68

トレース 23

トレースログ〔用語解説〕 68

は

バッチ業務サーバ〔用語解説〕 68

バッチ業務の OS 間での資産の継承 3

バッチ業務の構築のスピードアップ 3

バッチジョブ〔用語解説〕 68

バッチジョブ業務と実行順序の定義 51

バッチジョブ業務と実行順序の定義スケジュールの定義 52

バッチジョブ業務を開始する契機を登録する 52

バッチジョブの実行結果の一元管理 4

バッチジョブの実行結果の一元管理による運用性・保守性の向上 4

バッチ処理〔用語解説〕 68

ひ

引数〔用語解説〕 68

必要なディレクトリを作成する 23

標準エラー出力 (stderr)〔用語解説〕 68

標準出力 (stdout)〔用語解説〕 68

標準入力 (stdin)〔用語解説〕 68

ふ

ファイルシステムに関する注意事項 16

ファイルの割り当て〔用語解説〕 68

ブレークポイント〔用語解説〕 68

プログラム出力データファイル〔用語解説〕 68

へ

変数〔用語解説〕 68

り

リダイレクト〔用語解説〕 68

る

ルートジョブ〔用語解説〕 68

ろ

ローカルタイムの設定を確認する 16

ログ〔用語解説〕 68

わ

割り当て管理ファイル 54