

IBM MQSeries Workflow



定義機能の開始

バージョン 3.2.2

IBM MQSeries Workflow



定義機能の開始

バージョン 3.2.2

ご注意!

本書、および本書がサポートする製品をご使用になる前に、139ページの『付録C. 特記事項』にある一般的な情報を必ずお読みください。

本書は、IBM MQSeries Workflow (製品番号 5697-FM3) のバージョン 3、リリース 2、モディフィケーション 2、および新版において特に断りのない限り、それ以降のすべてのリリースおよびモディフィケーション・レベルに適用されません。

本書は SH88-7354-03 に対する新版です。

本マニュアルに関するご意見やご感想は、次の URL からお送りください。今後の参考にさせていただきます。

<http://www.ibm.com/jp/manuals/main/mail.html>

なお、日本 IBM 発行のマニュアルはインターネット経由でもご購入いただけます。詳しくは

<http://www.ibm.com/jp/manuals/> の「ご注文について」をご覧ください。

(URL は、変更になる場合があります)

原典：	SH12-6286-05 IBM MQSeries Workflow Getting Started with Buildtime Version 3.2.2
発行：	日本アイ・ビー・エム株式会社
担当：	ナショナル・ランゲージ・サポート

第1刷 2000.8

この文書では、平成明朝体™W3、平成明朝体™W9、平成角ゴシック体™W3、平成角ゴシック体™W5、および平成角ゴシック体™W7を使用しています。この(書体*)は、(財)日本規格協会と使用契約を締結し使用しているものです。フォントとして無断複製することは禁止されています。

注* 平成明朝体™W3、平成明朝体™W9、平成角ゴシック体™W3、
平成角ゴシック体™W5、平成角ゴシック体™W7

© Copyright International Business Machines Corporation 1993, 2000. All rights reserved.

Translation: © Copyright IBM Japan 2000

目次

本書について	v
本書の対象読者	v
本書で使用される表記規則と用語	v
追加情報の入手方法	vi

変更の要約	vii
MQSeries Workflow バージョン 3.2.2 の変更 点	vii
MQSeries Workflow バージョン 3.2.1 の変更 点	viii

第1部 定義機能を使ったモデル定義 . 1

第1章 定義機能の紹介	3
ワークフロー・モデルとは?	3
モデル定義に関係する人は?	4
モデル定義のステップとは?	5
MQSeries Workflow システムと定義機能	6
実行機能でのワークフロー・モデルの使用	7
定義機能と実行機能が協働する方法	8
定義機能と実行機能のデータベースの同期を 保つための指針	8

第2章 定義機能での作業	11
定義機能の開始	11
ヘルプの表示	12
定義機能インターフェースの使用	12
ビューおよびウィンドウの使用	13
メニュー・バーおよびツールバーの使用	14
ツール・パレットの使用およびカスタマイズ	15
ツリー・ビューの使用	17

第3章 プロセス・モデルの作成	19
スタッフの定義	19
スタッフ定義の計画	19
レベルの指定	21
人の定義	21
役割の定義	22
組織の定義	22
関連の表示	23

ネットワークの定義	23
プロセス・ダイアグラムの作成	24
プロセスの作成とそのプロパティの指定	25
データ構造の定義	35
プログラムの登録	36

第4章 スタッフの割り当てとプロセス・フローの定義	39
アクティビティのプロパティの指定	39
アクティビティへのスタッフの割り当て	39
動的スタッフ割り当ての指定	40
コネクターの論理の定義	43
プロセス・フローを制御するコネクター	43
データ・フローを制御するコネクター	44

第5章 ワークフロー・モデルの操作可能なプロセスへの変換	49
定義機能および実行機能でのワークフロー定義の使用	49
実行機能用のオブジェクトの状況の定義	49
ワークフロー・モデルの検証	51
定義機能からの搬出	60
定義機能へ搬入	61
実行機能搬出および搬入ユーティリティの使用	62
実行機能搬出 / 搬入ユーティリティの開始	63
搬出 / 搬入ユーティリティのオプション	66
搬出 / 搬入ユーティリティのエラー・コード	68
搬入の例	68
搬出の例	69
変換の例	70
MQSeries Workflow バージョン 3.1x のワークフロー・モデルをバージョン 3.2.2 で使う	70
FlowMark バージョン 2.3 FDL のワークフロー・モデルの使用	70

第2部 MQSeries Workflow の外部形式の使用 71

第6章 FDL ファイル内でのワークフロー情報の定義	73
構文図の読み方	73
FDL の構文規則	75
サイズ制限	76
名前と文字列の構文規則	76
条件の構文	81
第7章 FDL 定義	89
FDL ソース・ファイルの形式	89
FDL ソース・ファイル	89
トポロジ (Topology)	92
ドメイン (Domain)	92
システム・グループ (System Group)	92
システム (System)	93
TopologySetting	93
サーバー (Server)	98
ProgramExecutionAgent	99
QueueManager	99
スタッフ (Staff)	100
人 (Person)	100
役割 (Role)	101
組織 (Organization)	101
レベル (Level)	102
プロセス・モデル定義 (Process Modeling)	102
データ構造 (Data structure)	102
プログラム (Program)	103
プロセス (Process)	106
プロセス・カテゴリー (Process category)	114
ツール・セット (ToolSet)	114
共通変数	115
ScreenPosition	115
SymbolLayout	115
ContainerLayout	115
WindowLayout	116
ContainerInitial	116
BendPoints	116
カラー (Color)	116
ColorSetting	117
TextSettings	117
FontSettings	117
TimeStamp	117
TimeInterval	118
TimePeriod	118

TimeEvent	118
MessageLength	118
FullyQualifiedServerName	119

第3部 付録および後付け 121

付録A. スタッフ定義のモデル化についての詳細 123

スタッフ解決とは?	123
アクティビティーのためのスタッフ解決の定義	123
アクティビティーのプロパティを基にしたスタッフ解決	124
コンテナ・メンバーを基にしたスタッフ解決	127
プロセス・レベルでのスタッフ解決の定義	129
アクティビティーのプロセス定義の使用	130
継承するスタッフ解決の定義	130
プロセスのためのオプション	131
アクティビティーのためのオプション	132
スタッフ解決の評価のための規則	132
スタッフ解決のパフォーマンスの考慮	132
通知とは?	133
アクティビティーのプロパティを基にした通知	133
プロセスのプロパティを基にした通知	135
ネットワーク用に定義された通知	136

付録B. 定義機能データベースの再編成 . . . 137

定義機能と IBM DB2 ユニバーサル・データベース (DB2 Universal Database (R))	137
Microsoft Jet データベース・エンジンの使用	137

付録C. 特記事項 139

商標	141
----	-----

用語集 143

参照文献 151

MQSeries Workflow 資料	151
関連資料	151

索引 153

本書について

本書では、IBM MQSeries (R) Workflow (以降 MQSeries Workflow) の定義機能構成要素の概要を説明します。本書は、定義機能を使用してワークフロー・モデルを作成する方法について説明しています。また、定義機能で何ができるかを示す例を使って、モデル定義タスクを簡潔に紹介します。

本書の前半では、読者のビジネス・プロセス、およびそのプロセスを実行するのに必要なリソースを定義する方法について説明しています。

後半では、定義言語 (FDL) について説明しています。これは MQSeries Workflow で使用できます。MQSeries Workflow 交換形式 FDL を使用すると、既存のワークフロー定義を簡単に MQSeries Workflow に搬入したり搬出したりすることができます。

MQSeries Workflow についてより詳細に知りたい場合、または技術的な詳細について学習したい場合は、12ページの『ヘルプの表示』を参照してください。

本書には、MQSeries Workflow のすべてのコンポーネントの概説やインストールの方法は記載していません。MQSeries Workflow の他のコンポーネントについて説明する資料のリストは、151ページの『MQSeries Workflow 資料』を参照してください。

本書の対象読者

本書は、以下のことについてお知りになりたい場合にお読みください。

- 定義機能で何ができるか。
- 定義機能の使用方法。

MQSeries Workflow におけるワークフローの概念とアーキテクチャーに関する詳しい情報については、*IBM MQSeries Workflow: 概説およびアーキテクチャー*を参照してください。

本書で使用される表記規則と用語

規則

書名はイタリック体で示されています。

使用例

IBM MQSeries Workflow: 概説およびアーキテクチャー

メニュー・バー選択項目および押しボタンは太字 「OK」をクリックします。
体で示されています。

変数はイタリック体で示されています。重要な情報もイタリック体 で示されます。

プログラム *program name* がアクティビティーに割り当てられます。



この記号は、提案、重要なヒント、および実際的なテクニックを説明するときに付けられます。

追加情報の入手方法

次の MQSeries Workflow ホーム・ページをご覧ください。

<http://www.ibm.com/software/ts/mqseries/workflow>

関連資料のリストについては、151ページの『MQSeries Workflow 資料』を参照してください。

変更の要約

この節では、本書に加えられた、現在の版および旧版の主な改訂内容を列挙しています。

MQSeries Workflow バージョン 3.2.2 の変更点

定義機能の開始 このバージョンは、IBM MQSeries Workflow の新しいリリースに適用されます。

以下の章には、編集上の変更や技術的な更新が含まれています。

- 11ページの『定義機能の開始』には、構成 ID に関する追加情報があります。
- 44ページの『データ・コンテナ間でのデータのマッピング』には、編集上の変更があります。
- 47ページの『データ・コンテナ・メンバーのデフォルトの指定』には、編集上の変更があります。
- 61ページの『定義機能搬入の開始と使用』には、編集上の変更があります。
- 66ページの『搬出 / 搬入ユーティリティーのオプション』には、技術的な更新があります。この章は、新しい **f** オプションについて説明し、応答ファイル・オプションの使用に関する訂正情報を掲載しています。
- 73ページの『第6章 FDL ファイル内でのワークフロー情報の定義』には、技術的な更新があります。名前およびストリングの構文規則は拡張され、次のような新規の、または、変更された名前および規則が加えられました。
 - ModLevel
 - Priority
 - ProcessName
 - String
- 89ページの『FDL ソース・ファイルの形式』には、以下のセクションについての技術的な更新があります。
 - ModLevel の FDL ソース・ファイル
 - 処理アクション CREATE、REPLACE、UPDATE、DELETE のための、実行機能エクスポート / インポートに関するヒント

- QueueManager、Block、BlockSetting、ActivitySetting、ExplicitStaffAssignment、OrgAssignment、LevelAssignment、Notification の修正。
- 123ページの『付録A. スタッフ定義のモデル化についての詳細』は、ヒントを含む新しいセクションです。

MQSeries Workflow バージョン 3.2.1 の変更点

以下の章には、編集上の変更や技術的な更新が含まれています。

- 76ページの『名前と文字列の構文規則』
- 81ページの『条件の構文』
- 84ページの『終了条件および分岐条件の表記』
- 89ページの『FDL ソース・ファイルの形式』

第1部 定義機能を使ったモデル定義

第1章 定義機能の紹介	3
ワークフロー・モデルとは?	3
モデル定義に関係する人は?	4
モデル定義のステップとは?	5
MQSeries Workflow システムと定義機能	6
実行機能でのワークフロー・モデルの使用	7
定義機能と実行機能が協働する方法	8
定義機能と実行機能のデータベースの同期を 保つための指針	8
第2章 定義機能での作業	11
定義機能の開始	11
ヘルプの表示	12
定義機能インターフェースの使用	12
ビューおよびウィンドウの使用	13
メニュー・バーおよびツールバーの使用	14
ツール・パレットの使用およびカスタマイズ	15
ツリー・ビューの使用	17
第3章 プロセス・モデルの作成	19
スタッフの定義	19
スタッフ定義の計画	19
レベルの指定	21
人の定義	21
役割の定義	22
組織の定義	22
関連の表示	23
ネットワークの定義	23
プロセス・ダイアグラムの作成	24
プロセスの作成とそのプロパティの指定	25
プロセス・ダイアグラムの描画の開始	26
プロセス・ダイアグラムへのアクティビ ティの追加	26
プロセス・ダイアグラムの保管	28
プロセス・ダイアグラムの作図の指針	29
プロセス・ダイアグラム内のノードをコ ネクターと結合する	30
サブプロセス用のデータ・コンテナの 追加	31
プロセスのプロパティの指定	32

プロセス・ダイアグラムでのオブジェク トの移動	33
プロセス・ダイアグラムの部分コピー	33
プロセス・ダイアグラムの部分削除	34
データ構造の定義	35
デフォルト・データ構造	35
データ構造の定義	36
プログラムの登録	36
第4章 スタッフの割り当てとプロセス・フロ 一の定義	39
アクティビティのプロパティの指定	39
アクティビティへのスタッフの割り当て	39
動的スタッフ割り当ての指定	40
コネクターの論理の定義	43
プロセス・フローを制御するコネクター	43
データ・フローを制御するコネクター	44
データ・コンテナ間でのデータのマッ ピング	44
事前定義データ構造メンバーのマッピン グ	47
データ・コンテナ・メンバーのデフォ ルトの指定	47
第5章 ワークフロー・モデルの操作可能なプ ロセスへの変換	49
定義機能および実行機能でのワークフロー定 義の使用	49
実行機能用のオブジェクトの状況の定義	49
ワークフロー・モデルの検証	51
ワークフロー・モデルの検証に関する規 則	51
定義機能からの搬出	60
定義機能搬出の開始と使用	60
定義機能へ搬入	61
定義機能搬入の開始と使用	61
実行機能搬出および搬入ユーティリティの 使用	62
実行機能搬出 / 搬入ユーティリティの開 始	63
搬出 / 搬入ユーティリティのオプション	66

搬出 / 搬入ユーティリティーのエラー・コード	68
搬入の例	68
搬出の例	69
変換の例	70
MQSeries Workflow バージョン 3.1x のワークフロー・モデルをバージョン 3.2.2 で使う	70
FlowMark バージョン 2.3 FDL のワークフロー・モデルの使用	70

第1章 定義機能の紹介

ビジネス・プロセスの計画、計画変更、文書化、運用を手助けするのが、この MQSeries Workflow です。MQSeries Workflow は、日常の業務操作、計画と管理、および業務に合わせたアプリケーション設計に役立ちます。MQSeries Workflow を使うことによって、以下の作業を行うことができます。

- プロセスの定義と文書化
- より効率的な処理の実行
 - 処理している作業者の支援
 - 人間による決定を必要としないアクティビティーの完全自動化
 - ワークフローの管理

MQSeries Workflow はクライアント / サーバー・システムであり、定義機能はワークフロー・モデルでビジネス・プロセスを定義および文書化するための構成要素です。

通常、ビジネス・プロセスは、数多くのアクティビティーによって、あるいはさらに多くのアクティビティーを含むサブプロセスによって構成されます。プロセス内のさまざまなアクティビティーに対して、制御フローとデータ・フロー、およびプロセス内で使用したいアプリケーション・プログラムを指定します。

ワークフロー・モデルとは？

ワークフロー・モデルは、1 つまたは複数のビジネス・プロセスの完全な表現であり、関連するすべてのビジネス・アクティビティーによって構成されます。これには、ワークフロー参加者の定義と、ワークフローを実現するのに必要な IT 資源の定義も含まれています。

プロセスを定義するときは、MQSeries Workflow 定義機能のダイアログおよびグラフィカル・エディターを使用して、プロセス・ダイアグラムを作成します。ワークフロー・モデルを構築するには、3 つの主な構成要素のプロパティを定義する必要があります。4ページの図1 に、ワークフロー・モデルの 3 つの主な構成要素を示します。

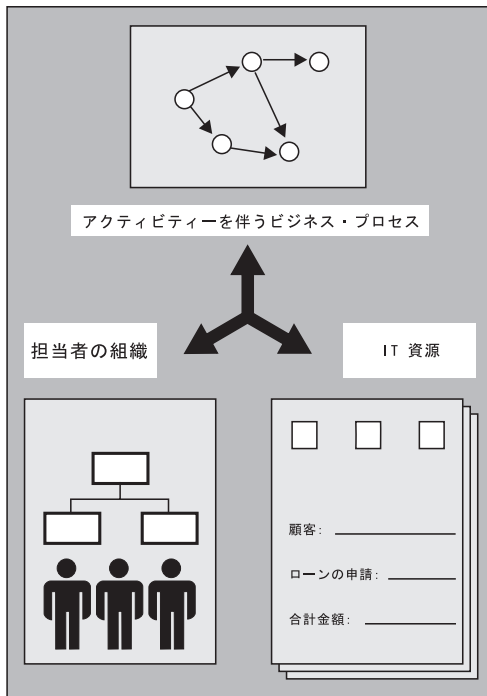


図1. ワークフロー・モデルの構築

- ・ アクティビティを伴うビジネス・プロセス
 MQSeries Workflow を使用すると、プロセスとそのアクティビティをプロセス・ダイアグラムに図示することができます。また、ワークフロー・モデルの背景となるプロセス論理を定義することもできます。
- ・ 担当者の組織
 組織のスタッフで、ビジネス活動を担当する人を指定します。
- ・ IT 資源
 MQSeries Workflow がビジネス・プロセスを実行するのに必要とする IT 資源を定義します。プロセスとそのアクティビティに使用したいデータおよびプログラムの定義を追加します。

モデル定義に関係する人は？

モデル定義には、いろいろな作業および技術が関係します。定義機能を使用すると、これらの作業を見分けられるようにシステムをセットアップできます。異なる作業を複数のユーザーが担当したり、1人のユーザーが複数の作業を実行したりすることができます。

システム管理者

システム管理者は、第 1 人者として、MQSeries Workflow システムに存在しています。システム管理者は、他のスタッフ・メンバーの初期定義を行います。ワークフロー・モデルが確立された後は、ワークフロー・モデルの保守と、実行しているプロセスの監視を担当します。

スタッフを定義するユーザー

システム管理者により、データベースのスタッフ・メンバーの定義を作成および変更することを許可されているユーザーです。

モデルをプロセスするユーザー

プロセス・モデルを構築および検証することを許可されているユーザーです。これらのプロセス・モデルは、実行時のプロセスの実行方法を定義します。

IT タスクが許可されているユーザー

MQSeries Workflow で使用するプログラムを、設計および定義することを許可されているユーザーです。

モデル定義のステップとは？

モデル定義プロセスの各ステップは、互いに依存しています。

以下のステップを示された手順で完了すると、各ステップの前提条件が満たされます。

1. スタッフ・メンバーの組織を定義します。これには、その組織で必要な役割およびレベルを定義することが含まれます。
2. ドメインのネットワーク特性と、サーバー構成要素を定義します。
3. プロセスに必要なデータ構造と、プロセス内およびプログラム内のアクティビティを定義します。
4. プロセス内のアクティビティで必要なアプリケーション・プログラムやツールを登録します。
5. 各アクティビティおよびブロックを、制御とデータの流れを判別するコネクタを含めて表示し、プロセスのダイアグラムを描きます。プロセスのプロパティを指定します。
6. プロセス・ダイアグラムの背景となる論理を詳細に定義します。
 - 各アクティビティについて、その開始条件と終了条件、およびアクティビティを実行するのに必要な人、データ構造、およびプログラムを指定します。

- ダイアグラムの制御コネクタのそれぞれに、任意で分岐条件を指定します。この分岐どおりに制御が流れるにはこの条件が必要となります。
- ダイアグラムのデータ・コネクタのそれぞれに、1つのアクティビティの出力コンテナにあるデータが、別のアクティビティの入力コンテナにマップされる方法を指定します。

ワークフロー定義を定義機能から実行機能へ転送するには、49ページの『第5章 ワークフロー・モデルの操作可能なプロセスへの変換』にしたがって、まず定義機能搬出ユーティリティを使用し、それから実行機能搬入ユーティリティを使用して FDL を実行機能機能データベースに搬入します。ワークフロー・モデルは 62ページの『実行機能搬出および搬入ユーティリティの使用』にしたがって実行機能搬入ユーティリティを使用し、その実行機能プロセス・テンプレートに変換します。ワークフロー・モデルを検証する方法についての詳細は、51ページの『ワークフロー・モデルの検証』を参照してください。

MQSeries Workflow システムと定義機能

定義機能は MQSeries Workflow システムの一部であり、プロセス・モデルを作成するためのグラフィカル・エディターが用意されています。定義機能は、専用のリレーショナル・データベースを使用して、プロセス・モデルについての情報を保管します。7ページの図2 に、定義機能でのシステム・アーキテクチャおよびそのデータベースを示します。

3層のアーキテクチャ

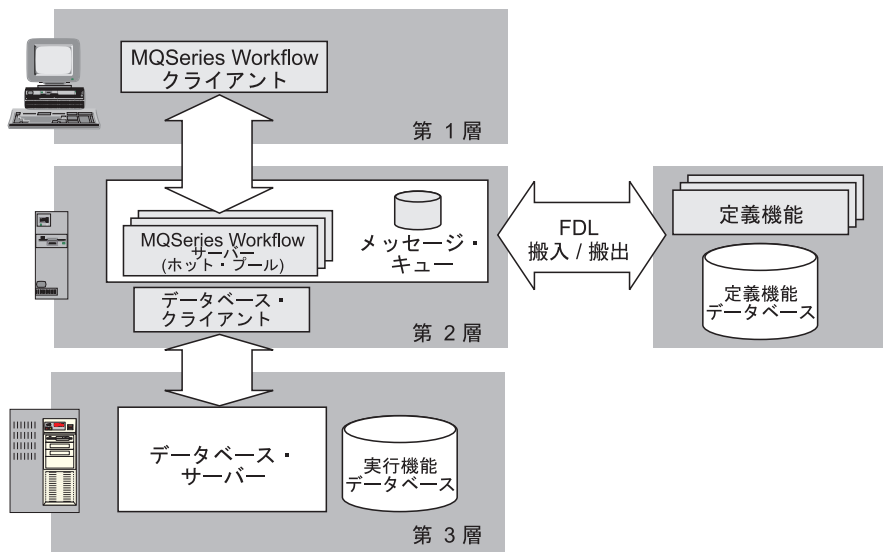


図2. システム・アーキテクチャ

MQSeries Workflow のアーキテクチャについての詳細は、*IBM MQSeries Workflow: 概説およびアーキテクチャ* を参照してください。

新しいプロセス・モデルを定義するのに定義機能を使用する代わりに、MQSeries Workflow 定義言語 (FDL) ですでに定義されているモデル情報を使用することができます。 FDL ファイルを定義機能データベースに搬入できます。たとえば、これは、プロセス・モデルの交換形式として FDL を提供するビジネス・モデル化ツールを使用する場合に適用されます。また、これらの定義を MQSeries Workflow 定義機能に直接搬入することもできます。 FDL の使用方法についての詳細は、49ページの『定義機能および実行機能でのワークフロー定義の使用』を参照してください。

実行機能でのワークフロー・モデルの使用

定義機能でワークフロー・モデルを作成しこれをテストした後は、49ページの『第5章 ワークフロー・モデルの操作可能なプロセスへの変換』で説明されているように、これを定義機能から搬出し、実行機能に搬入します。これは、プロセス・テンプレート と呼ばれます。 実行機能で、プロセスのすべてのインスタンス について、MQSeries Workflow のサーバー構成要素は、プロセスをナビゲートします。 MQSeries Workflow は、プロセス・モデル情報を使用し

て、作業を適切な人に正しい順序で割り当てていきます。MQSeries Workflow は定義したプログラムを開始し、プロセス実行履歴を保持し、回復および再始動プロシージャを提供します。

実行する必要があるアクティビティーは、割り当てたスタッフ・メンバーのMQSeries Workflow クライアントのワークリストに表示されます。たとえば、スタッフ・メンバーがプログラム・アクティビティーを選択すると、そのプログラムは、プロセス・モデルで定義した情報に従って開始されます。

定義機能と実行機能が協働する方法

7ページの図2 で示されているように、定義機能は固有のデータベースを使用します。このデータベースは、実行機能で使用されるメイン・データベースから独立しています。これには、次のような利点があります。

- モデル定義担当者は、メイン・データベース、つまり実行機能データベースから独立して作業できます。
- モデル定義データベースは、実行中のプロセスにパフォーマンスの影響を与えずに使用できます。
- メイン・データベースには異なるオペレーティング・システムを使用できます。たとえば、強力な AIX (R) サーバーを使用できます。さらに、定義機能では、Windows NT または Windows 9x を使用することができます。定義機能にモバイル・システムを使用することもできます。
- データベース表を特定の目的に合わせて最適化できます。定義機能データベースは、長時間の実行トランザクションをサポートしなければならないのに対し、実行機能データベースは、ワークフロー実行のトランザクション・パターンに対応するように最適化することができます。

プロセスを実行機能で使用する準備ができたなら、モデル情報を定義機能データベースから FDL ファイルに搬出しなければなりません。それから FDL ファイルを実行機能データベースに搬入できます。詳細については、49ページの『第5章 ワークフロー・モデルの操作可能なプロセスへの変換』を参照してください。



MQSeries Workflow の 定義機能と実行機能との間でプロセス・モデルを交換するには、定義機能および実行機能の搬出および搬入ユーティリティを使用しなければなりません。以下のガイドラインに従えば、データベースの整合性を保つことができます。

定義機能と実行機能のデータベースの同期を保つための指針

データベースの整合性を保つために、以下の指針に従ってください。

- MQSeries Workflow ドメインに 1 つの定義機能データベースを作成します。
- 定義機能を使用してワークフロー・モデルに必要なすべての定義を作成します。
- 定義機能で**削除マーク**を使用して、実行機能データベースでのデータの削除に備えます。
- 定義機能から FDL ファイルにデータを搬出します。
定義機能では、適切な FDL キーワードを使用すると、**新規、不確定、更新**、または**削除マーク付き**などのデータを選択して搬出できます。
- 実行機能搬入ユーティリティを使用して、FDL を実行機能データベースに搬入します。
- FDL ファイルを実行機能に搬入した後は、次のようにして、定義機能データベースを更新します。
 - **削除マーク付き**の項目を削除します。
 - **新規**または**更新**のマークを付けた項目、および**不確定**のマークを付けた項目のマークを取り除きます (デフォルトにリセットする)。

定義機能では、シンボルを使用してオブジェクト状況 を表示します。これにより、実行機能でのワークフロー・モデルの項目の状況を示します。詳細については、49ページの『実行機能用のオブジェクトの状況の定義』を参照してください。

- データベースの同期をとるには、関連した定義機能データベースから得られたもの以外の FDL ファイルは、実行機能データベースに搬入しないでください。

定義機能および実行機能データベースは自動的に同期化されるわけではないため、クライアントから、または API プログラミングを介して変更すると、この 2 つのデータベース間で不整合になる可能性があることに注意しなければなりません。その種の変更には、以下のものが含まれます。

- ユーザーのパスワードの変更。
- 人に不在フラグを付けたりそれをリセットしたりすること。不在フラグは、そのユーザーがクライアントにログオンすると、自動的にリセットされません。
- 代行者の情報の変更。

実行機能データベースにワークフロー・モデルを搬入するには、49ページの『第5章 ワークフロー・モデルの操作可能なプロセスへの変換』に説明されているように、実行機能搬入ユーティリティを使用しなければなりません。

第2章 定義機能での作業

この章では、定義機能を開始し、グラフィカル・インターフェースを使用して作業する方法について説明します。ここでは、システム管理者が *IBM MQSeries Workflow*: インストールの手引き に記載されている手順に従って定義機能インストールのセットアップを完了していることが前提とされています。

定義機能の開始

定義機能を開始する前に、MQSeries Workflow のインストールを担当している管理者に問い合わせ、システムのセットアップ方法を確認してください。

定義機能を開始するには、次のようにします。

1. タスクバーで「スタート」をクリックします。
2. 「プログラム」を選択し、「MQSeries Workflow」を選択します。
3. **MQSeries Workflow 定義機能 ID** をクリックします。ここで、ID は構成 ID です。FMC はデフォルト構成 ID です。

複数の定義機能インスタンスが使用できます。各インスタンスには固有のデータベースがあり、これはローカル・データベースにも、リモート・データベースにもなり得ます。複数インスタンスがあれば、始動させたい定義機能インスタンスを選択できます。使用可能なインスタンスを見分けるために、構成 ID が表示されます。始動させたいインスタンスをクリックします。

MQSeries Workflow の構成方法の詳細については、*IBM MQSeries Workflow*: インストールの手引き を参照してください。

管理者が定義機能を開始するためのショートカットを作成していた場合には、そのショートカットをクリックします。

注:

- 定義機能データベースにも定義されているユーザー ID で Windows オペレーティング・システムにログオンすると、定義機能に自動的にログオンします。そのため、ログオン・ウィンドウは表示されません。
- 定義機能データベースに定義されていないユーザー ID で Windows オペレーティング・システムにログオンすると、ログオン・ウィンドウが表示されます。そこで有効な定義機能ユーザー ID を指定することができます。

すでにログオンしているものとは別のユーザー ID でログオンする方法についての詳細は、オンライン・ヘルプの**ログオン・トピック**を参照してください。

ヘルプの表示

定義機能の使用時には、オンライン・ヘルプが主な情報源になります。ウィンドウに関する情報は、「**目次 (Contents)**」セクション、および「**索引 (Index)**」または「**検索 (Find)**」で検出できます。



特定のフィールドに関するヘルプを表示するには、次のようにします。

- 疑問符 (?) ボタンをクリックしてから、フィールドをクリックします。

すべてのヘルプ・トピックを表示するには、次のようにします。

1. 「**スタート**」ボタンをクリックします。
2. 「**プログラム**」をポイントします。
3. 「**エクスプローラ**」をクリックします。

MQSeries Workflow がインストールされているディレクトリーには、**fmcbhenu.hlp** という名前のヘルプ・ファイルがあります。ここで、*enu* は米国英語版を表します。他の言語省略記号については、*IBM MQSeries Workflow: インストールの手引き* を参照してください。

4. ヘルプ・ファイルをダブルクリックして、定義機能のオンライン・ヘルプ・トピックを表示します。

定義機能インターフェースの使用

定義機能を最初に開始すると、13ページの図3 に示されているような定義機能ウィンドウが表示されます。ただし、ダイアグラムは表示されません。

定義機能ウィンドウの左には、ツリー・ビュー があり、ワークフロー・モデルに属するすべてのオブジェクトが表示されます。ツリー・ビューの上部のタブを使用すると、別のツリーに素早く切り替えられます。このタブで、プロセス、スタッフ、ネットワーク、およびインプリメンテーション のオブジェクト・ツリーを表示することができます。

定義機能ウィンドウの右側には作業域 があります。ここは、ワークフロー要素のビューを表示するために使用されます。ワークフロー要素のビューは、プロセスのダイアグラム・ビューや、選択されたオブジェクトに定義できるプロパティなどです。

定義機能ウィンドウの下部には、状況バー があります。状況バーには、使用しているデータベースの名前やユーザー ID などの情報が表示されます。

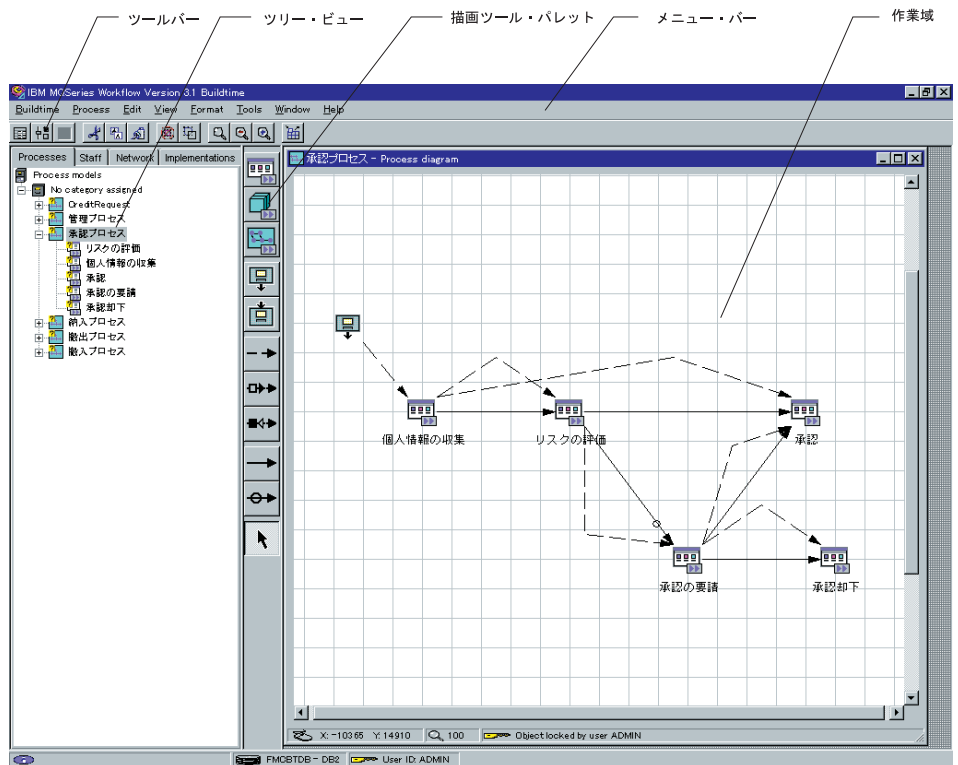


図3. 定義機能ユーザー・インターフェース

ビューおよびウィンドウの使用

ツリー・ビュー内のオブジェクトを選択する場合、そのオブジェクトを異なるタイプのビューで選択することもできます。ビューを選択するには、以下のようになります。

- ツリー・ビューで要素を右クリックします。たとえば、クレジット要求 (*CreditRequest*) というプロセスなどです。

- サブメニューで「**ダイアグラム**」をクリックすると、選択されたプロセスのプロセス・ダイアグラムが表示されます。

ツリーで選択する要素により、3つの異なるタイプのビューがあります。

プロパティ

選択された要素の定義を含むプロパティ・ページがあります。これらのページには、各ページを示すタブが含まれています。たとえば、ワークフロー・モデルでユーザーのプロパティを定義し、表示することができます。

詳細 このビューは、他の要素のコンテナである要素に使用できます。たとえば、詳細ビューで、あるプロセス・カテゴリーに属するプロセスの詳細を表示できます。詳細は、スプレッドシート形式で表示されます。

定義機能オブジェクトの独自のビューを作成することもできます。詳細ビュー設計機能 (**Details View Designer**) を使用するには、メニュー・バーで「**ツール**」をクリックし、それから「**Details View Designer**」をクリックします。次にオブジェクト・タイプを選択し、望みのプロパティを選択できます。Details View Designer は、定義機能データベースに保管されているオブジェクトの照会を開始します。

ダイアグラム

ダイアグラム・ビューは、プロセスをグラフィック表示します。ダイアグラム・エディターを使用して、プロセス・ダイアグラムを作成または変更できます。

メニュー・バーおよびツールバーの使用

ツリー・ビューに加えて、定義機能ウィンドウにはメニュー・バー およびツールバー があります。15ページの図4 に表示されているように、メニュー・バーから項目を選択したり、ツールバーを使用することができます。メニュー・バーおよびツールバーの項目は、選択したビューまたはウィンドウに関連して表示されます。たとえば、スタッフ・タブをクリックすると、スタッフ・ツリーがオープンして、スタッフ定義を指定するのに必要な項目がメニュー・バーに入ります。したがって、プロセスのメニュー項目はありません。

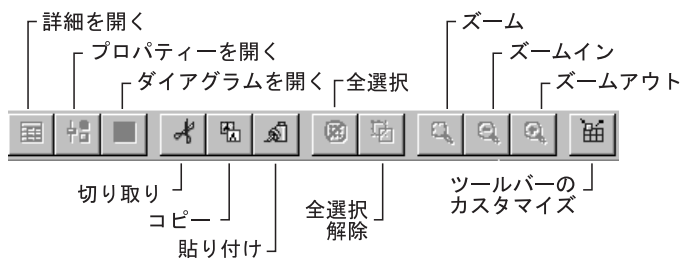


図4. ツールバーの使用

ツールバーの項目についての詳細は、オンライン・ヘルプを参照してください。

ツール・パレットの使用およびカスタマイズ

選択したプロセスのダイアグラム・ビューをオープンすると、プロセス・ダイアグラムが右の区画に表示されます。13ページの図3では、右の区画にダイアグラム・ビューが表示されています。ツリー・ビューとダイアグラムとの間には、16ページの図5に示されているように、描画ツール・パレット (略して「ツール・パレット」) を表示することができます。描画ツール・パレットを表示するには、「ビュー」メニューから「描画ツール (Drawing Tools)」をクリックし、それから「表示」を選択します。ツール・パレットを隠したい場合には、「隠す (Hide)」をクリックします。どの機能を選択したかがチェック・マークで示されます。

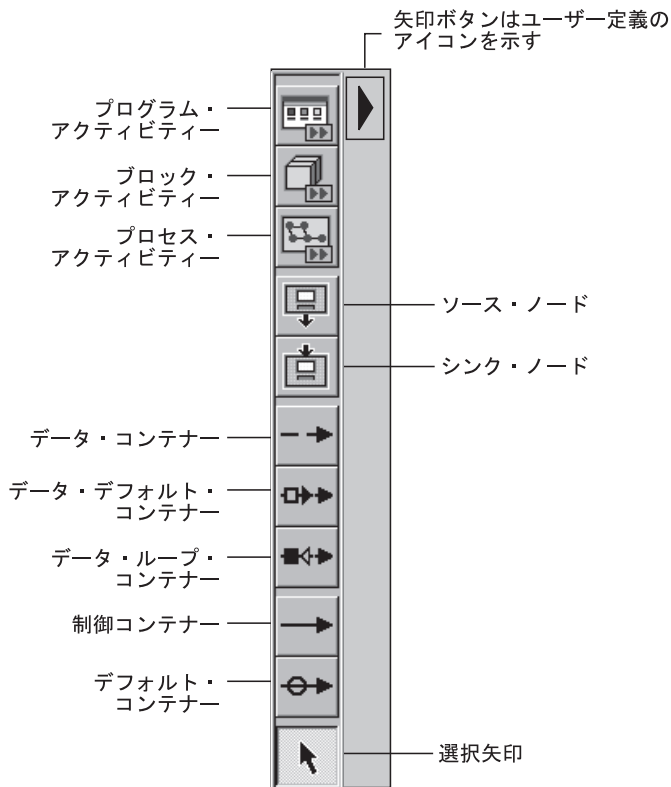


図 5. ツール・パレットの使用



自分で作成したアイコンをツール・パレットに使用したい場合には、「カスタマイズ」をクリックします。インストールのセットアップの仕方によっては、ここでユーザー定義のアイコンを選択することができます。矢印ボタンは、使用可能なユーザー定義のアイコンがあることを示しています。デフォルトで提供されているアイコンの代わりにこれらのアイコンを使用することができます。

プロセス・ダイアグラムの描画に使用する独自のアイコンをインストールする方法については、*IBM MQSeries Workflow: インストールの手引き* を参照してください。適切なインストール・パスを使用して、実行機能のユーザー定義アイコンも確実にインストールしてください。

ツール・パレットにアイコンを追加する方法については、オンライン・ヘルプを参照してください。

ツリー・ビューの使用

適切なタブを選択すると、作業対象のツリー・ビューを選択できます。以下の作業の対象を選択できます。

プロセス

プロセスのツリー・ビューは、定義機能データベースに保管されるすべてのカテゴリ、プロセス・モデル、およびそれらのアクティビティを表示します。プロセスは、割り当てるカテゴリに従ってソートされます。プロセスにカテゴリがない場合、それらのプロセスは、「カテゴリの割り当てなし (*No category assigned*)」の下に表示されます。

スタッフ

スタッフのツリー・ビューは、組織に定義したすべての要素、つまり、人、役割、組織、およびレベルを表示します。

ネットワーク

ネットワーク定義のツリー・ビューは、階層の頂点にドメイン・ネームがあり、システム構成要素が階層状に表示されます。MQSeries Workflow のシステム構造に属するシステム・グループ、システム、およびサーバーがツリー・ビューに表示されます。

インプリメンテーション

インプリメンテーションのツリー・ビューでは、ワークフローに定義するすべてのデータ構造およびプログラムを表示します。

第3章 プロセス・モデルの作成

この章では、スタッフ、ネットワーク・プロパティ、データ構造、およびプログラムを定義する方法について説明します。また、プロセス・ダイアグラムを描画する方法についても説明します。

ワークフロー・モデルの構成要素を定義するのに必要なプロセス論理については、39ページの『第4章 スタッフの割り当てとプロセス・フローの定義』を参照してください。

プロセス・モデルを実行時に使用するとき MQSeries Workflow が検査する事柄についての詳細は、51ページの『ワークフロー・モデルの検証』を参照してください。

スタッフの定義

ワークフロー・モデル内のすべてのプロセスおよびアクティビティは、ユーザー ID で識別される 1 人または複数の担当者に関連付けられなければなりません。つまり、ワークフロー・モデルで言及されるすべての担当者は、すでに定義機能データベースに定義されていなければなりません。ただし、アクティビティを役割に割り当てることができます。つまり、アクティビティを明示的に個人に割り当てる必要はないということです。ワークフロー・モデルまたは個々のプロセスを作成する人もみな定義されていなければなりません。スタッフを動的にまたは個別に割り当てる方法についての詳細は、39ページの『第4章 スタッフの割り当てとプロセス・フローの定義』を参照してください。

ツリー・ビューで「スタッフ」タブを選択すると、20ページの図6のように、スタッフ・オブジェクトが表示されます。ツリー・ビューで示されているオブジェクトのプロパティについて作業したり、オブジェクトを追加または削除することができます。

スタッフ定義の計画

企業の初期スタッフ定義を行うのは、おそらくシステム管理者です。

最も単純なプロセスでは、個々のスタッフ・メンバーを定義するだけで十分な場合があります。しかし、作成するプロセスにおいて、スタッフへのアクティビティーの割り当てを柔軟に行えるようにしたい場合は、以下のスタッフ定義を行うこともできます。

- 役割
- 組織
- レベル

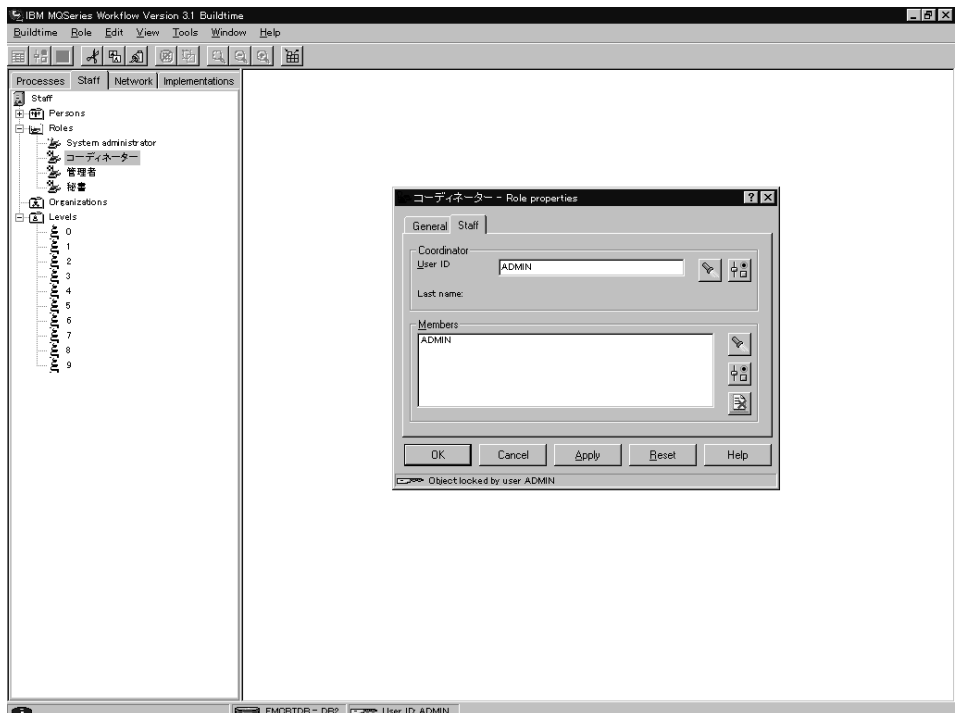


図 6. スタッフ・ツリーの使用

これらのうち 1 つまたはいくつかをスタッフ・メンバーの定義と関連付けることによって、以下のことが可能になります。

- アクティビティーを割り当てることのできる、人のグループを作成する。
- レベル、組織、または役割などに関連した特定の基準に適合する人に、アクティビティーを動的に割り当てる。



次の順序でスタッフを定義します。

- レベル
- 人
- 役割および組織
- 人と役割の関係
- 人と組織の関係

権限考慮事項についての詳細は、*IBM MQSeries Workflow: プログラミングの手引き* を参照してください。

レベルの指定

人の区別を行うために、レベルをそれぞれの人に割り当てます。これらのレベル分けは、適当な基準を設けて行うことができます。たとえば、最長の経験者、あるいは最高の技術をもった人に対しては、最高のレベルを割り当てます。

これで、アクティビティーに対する動的割り当てを行うときに、候補のフィルターを行う基準としてレベルを使用することができます。

「スタッフ」ツリーでは、「レベル」ツリーで 10 のレベルを表示できます。人は 0 から 9 までのレベルで割り当てます。これらのレベルは事前定義されており、名前および説明のみ変更できます。レベルの作成または削除を行うことはできません。

レベルのプロパティをオープンするには、スタッフ・ツリー・ビューで、レベル・オブジェクトを右マウス・ボタンでクリックし、それから「プロパティ」をクリックします。「一般」ページがオープンします。ここで、レベルに名前を割り当て、そのレベルの説明を入力します。

人の定義

プロセスに組み込む人を MQSeries Workflow に識別させるには、「人」ツリーを使用してください。「人」ツリーは、スタッフを表す担当者オブジェクトを表示します。スタッフのメンバーに、MQSeries Workflow で異なる機能を実行する許可を与えることができます。これらは、定義機能から搬出したり、実行機能に搬入したりする際に、実行機能で有効です。担当者オブジェクトを作成する方法についての詳細は、オンライン・ヘルプを参照してください。

役割の定義

「役割」ツリーは、企業内での役割を示すのに使用される役割オブジェクトを表示します。

MQSeries Workflow での役割は、1 人の人、あるいは一まとまりの人たちが共通に持っている機能または能力のことを指します。たとえば、これは、特定の作業グループのメンバーにすることができます。1 人の人が多くの役割を持っている場合もあります。多くの人が同じ役割を持つ場合もあります。アクティビティーを役割に割り当てると、その役割を持つすべての人が、同じアクティビティーを実行機能のワーク・リストで受け取ります。そして、そのうちのどれでも、そのアクティビティーを実行することができます。

企業内のスタッフの役割を定義するとき、それらの役割のコーディネーターも定義することができます。たとえば、MQSeries Workflow にチームを定義する場合は、クレジット・スタッフと呼ばれる 1 つの役割について 1 人のコーディネーターを定義することができます。このクレジット・スタッフと呼ばれる役割のメンバーが、そのチームのメンバーであり、コーディネーターはそのチームのリーダーになります。コーディネーターがそれぞれの役割を持つ人のワーク・リストにアクセスできるよう、許可することができます。すると、実行時に、コーディネーターは、チーム・クレジット・スタッフのメンバーの間でアクティビティーを分散させることができます。

MQSeries Workflow は、定義機能データベースで 1 つの役割 (システム管理者) を事前定義しています。システム管理者の役割にはすべての MQSeries Workflow の許可が含まれます。この役割は常に少なくとも 1 人の人に割り当てられなければなりません。ただし、割り当ては、ある人から別の人に変更することができます。

役割オブジェクトを作成するには、スタッフ・ツリー・ビューから、「役割」オブジェクトを右ボタンでクリックし、それから「**新規の役割 (New Role)**」をクリックします。役割のプロパティーが表示されます。新しい役割のプロパティーを定義できます。入力フィールドについての詳細は、オンライン・ヘルプを参照してください。

組織の定義

MQSeries Workflow における組織とは、企業の構造を記述する管理単位です。MQSeries Workflow では、組織は、階層状に配置されます。1 つの組織には 1 つの上位組織しかありませんが、下位組織はいくつでも持つことができます。1 人の人は 1 つの組織のメンバーにしかありません。それぞれの組織には、管理者として割り当てられている人がいなければなりません。

「組織」ツリーは、これらの管理単位を表すために使用される組織オブジェクトを表示します。

組織オブジェクトを作成するには、スタッフ・ツリー・ビューから、「組織」オブジェクトを右ボタンでクリックし、それから「**新規の組織 (New Organization)**」をクリックします。組織のプロパティが表示されます。新しい組織のプロパティを定義できます。入力フィールドについての詳細は、オンライン・ヘルプを参照してください。

関連の表示

定義機能データベースで、人と役割や、他のオブジェクトとの関連を表示したい場合には、**関連ビューアー** をオープンすることができます。

特定の役割にだれが割り当てられているかを表示するには、「**ツール**」メニューから、「**関連ビューアー (Relations Viewer)**」をクリックします。オブジェクト・タイプ (たとえば「役割」) を選択します。特定のオブジェクト (たとえば「コーディネーター」) を選択します。ユーザー ID を表示するには、「**割り当て先 (is assigned to)**」をクリックします。

ネットワークの定義

ワークフローにおいて完全操作性を実現し、すべてのアクティビティを自動実行できるようにするには、MQSeries Workflow ネットワークのためのプロパティを指定しなければなりません。ネットワーク・ツリーは、階層状に配置されます。最高レベルには、MQSeries Workflow ネットワークのドメインが表示されます。

MQSeries Workflow に定義または搬入するワークフロー・モデルは、ドメインについて有効です。これには、スタッフ、データ構造、プログラム、およびプロセスの定義すべてが含まれます。この最高レベルでは、MQSeries Workflow インストール・システムの性質を指定するプロパティを定義できます。最高レベルで作成する定義は、すべての下位レベルに継承されます。下位レベルで異なる定義を望む場合には、それを明示的に定義し、その定義はそのレベルで有効になります。

MQSeries Workflow のアーキテクチャーについての詳細は、*IBM MQSeries Workflow: 概説およびアーキテクチャー* を参照してください。

ネットワーク・オブジェクトのプロパティ (たとえば「システム・グループ」) を操作するには、次のようにしてください。

1. 「ネットワーク」ツリー・ビューで、「システム・グループ」オブジェクトを右クリックします。
2. 「プロパティ」をクリックします。
「システム・グループのプロパティ」ウィンドウが表示されます。
3. これらの設定値を入力した後で、「了解」をクリックして変更を確認します。

入力フィールドおよび名前の構文についての詳細は、オンライン・ヘルプを参照してください。

プロセス・ダイアグラムの作成

さまざまなタイプのアクティビティーを含むプロセス・モデルのダイアグラムを描画することができます。通常、プロセスを構成する個々のアクティビティーが数多くあります。プロセスには、さらに多くのアクティビティーを含むサブプロセスやブロックも入れられます。プロセスおよびブロック・アクティビティーの詳細については、26ページの『プロセス・ダイアグラムへのアクティビティーの追加』を参照してください。プロセスのアクティビティーの指定に加えて、制御フローおよびデータ・フローも指定する必要があります。プログラム・アクティビティーには、プロセス内で使用したいアプリケーション・プログラムを指定しなければなりません。ワークフローを完成するのに必要なスタッフおよびネットワーク定義も、プロセス・モデルの一部です。



はじめに、実行するアクティビティーの順序も含めたプロセス・モデル全体の作図を行って、概要をつかむことをお勧めします。

プロセスとそのアクティビティーのプロパティ、および制御フローとデータ・フローのプロパティは、ダイアグラムを作成した後で指定できます。詳細については、39ページの『第4章 スタッフの割り当てとプロセス・フローの定義』およびオンライン・ヘルプを参照してください。

ワークフローのモデル化のステップがどのように相互に依存し合っているかについての詳細は、5ページの『モデル定義のステップとは？』を参照してください。

アクティビティーの永久ループの作成などのモデル定義エラーを回避するために、MQSeries Workflow は、誘導グラフを使用してプロセス・ダイアグラムを作成します。アクティビティーのグループを接続して、制御またはデータの循環する流れを形成することはできません。しかし、条件が合うまで繰り返し実行されるアクティビティーの終了条件を定義したり、条件が合うまで一連の

アクティビティーが繰り返し実行されるブロックの終了条件を定義することはできます。また、終了条件が適合しないときに、出力コンテナのデータを入力コンテナのデータにマップさせるアクティビティーまたはブロックのために、データ・コネクタをループの形で作成することもできます。したがって、繰り返されるアクティビティーまたはブロックは、同じアクティビティーまたはブロックの前の処理中に生成されたデータへのアクセス権を持ちます。終了条件についての詳細は、43ページの『プロセス・フローを制御するコネクタ』を参照してください。

プロセスの作成とそのプロパティの指定

新しいプロセスを作成するには、そのプロセスのプロパティを定義し、プロセス・ダイアグラムを描画しなければなりません。特定の 카테고리に従ってプロセスをグループ化したい場合には、カテゴリを定義することができます。これはユーザーが実行機能の許可権限を管理するための助けになります。たとえば、銀行の環境で、ローン申請用のプロセスを定義しようとしているときに、これらのプロセスのカテゴリを定義することができます。誰に特定のプロセス・カテゴリの権限を与えるかを定義し、特定のカテゴリのプロセスを開始する権限を持つ人々の数を制限したりすることができます。

新規のカテゴリを作成するには、次のようにします。

1. ツリー・ビューで、「プロセス・モデル」を右クリックします。
2. 「新規のカテゴリ」をクリックします。
3. 「カテゴリのプロパティ」ダイアログ・ボックスにカテゴリの名前を入力します。
4. 「了解」をクリックして、新規のカテゴリを保管します。

作成したばかりのカテゴリのもとに新しいプロセスを作成するには、次のようにします。

1. 「銀行業務 (Banking)」などのカテゴリを右クリックします。
2. 「新規のプロセス」をクリックします。
3. この新しいプロセスの名前を入力します。
4. 「了解」をクリックして、新しいプロセス名を保管します。

これで、プロセスの欠落したプロパティを追加する前に、プロセス・ダイアグラムを描画できます。プロセスにプロパティを追加する方法についての詳細は、32ページの『プロセスのプロパティの指定』にあります。

プロセス・ダイアグラムの描画の開始

新しいプロセスのダイアグラムを描画するには、次のようにします。

1. ツリー・ビューで、作成したばかりのプロセス名を右クリックします。
2. 「ダイアグラム」をクリックします。

プロセス・ダイアグラムが作業域にオープンします。

プロセス・ダイアグラムへのアクティビティーの追加

27ページの図7 は、新しく作成されたプロセスのダイアグラム・ビューを示しています。これには、プログラム・アクティビティーと制御コネクターが含まれ、これらのアクティビティーの可能な順序を示します。プロセス・ダイアグラムは、ノード (アクティビティー、ソース、およびシンク・コンテナ) と、方向性をもったコネクター (制御コネクター、データ・コネクター) で構成されるプロセスを表現したものです。

ツール・パレットは、プロセス・ダイアグラムですべての要素を描画するためのオブジェクトを提供します。ツール・パレットは、オンライン・ヘルプで説明されているように、カスタマイズできます。



各オブジェクトは、異なる作業および状況に適合されるものなので、ダイアグラムを作成する前に、モデルを詳細に計画しておくことが重要です。

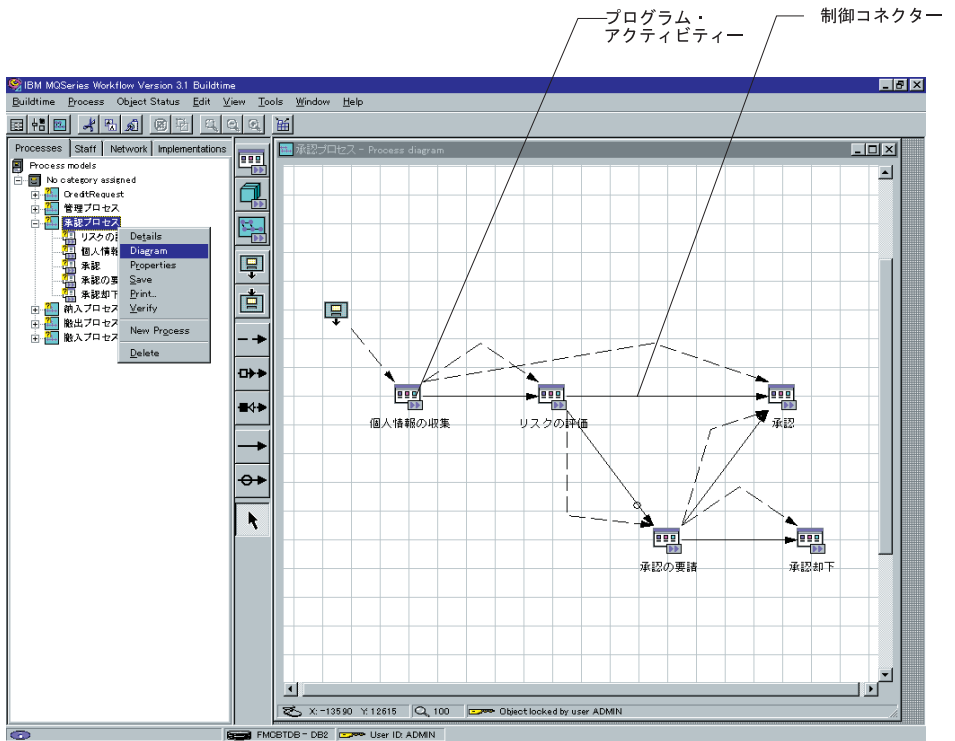


図7. アクティビティおよび制御フローの描画

以下のアイコンを使用して、アクティビティをモデル化できます。

表1. アクティビティ用のアイコン



プログラム・アクティビティには、プログラムが割り当てられています。このプログラムは、実行機能でアクティビティを開始したときに呼び出されます。プログラムが終了すると、プログラム・アクティビティの終了条件が評価されます。終了条件の評価の結果によって、アクティビティは終了するか、または「作動可能」状況になります。手動による終了がそのアクティビティに指定されている場合、アクティビティを開始する人が、そのアクティビティの終了を確認しなければなりません。

表 1. アクティビティー用のアイコン (続き)



プロセス・アクティビティーは、実行機能のワーク・リストから開始する他のプロセス (サブプロセス) を定義します。プロセスは、アクティビティーを開始したときに呼び出されます。呼び出されたプロセスが終了すると、プロセス・アクティビティーの終了条件が評価されます。評価の結果によって、プロセス・アクティビティーは、終了するか、または再開します。



ブロック・アクティビティーは、終了条件を満たすまで繰り返される一連のアクティビティーの集合を定義します。ブロックは `do-until` ループを定義するときに使用します。また、複合モデルでいくつかのアクティビティーを 1 つにまとめるためにも使用できます。

終了条件についての詳細は、43ページの『プロセス・フローを制御するコネクター』を参照してください。

これらのオブジェクトをプロセス・ダイアグラムに追加するときは、以下の手順で行ってください。

1. ダイアグラムに追加したいアクティビティーまたはブロックを示す、ツール・パレット内のシンボルをクリックします。
2. マウス・ポインターを、ダイアグラム内でノードを置きたい位置まで移動します。

作図エリアで、マウス・ポインターは、パレットでユーザーが選択したシンボルの形に変化します。

3. 描画エリアで、ダイアグラムにノードを置きたい位置でクリックします。



同じタイプのノードをダイアグラムに追加するときは、マウス・ポインターを別の位置に持っていき、1 度クリックするだけでできます。別のタイプのノードまたはコネクターを作図するときは、そのシンボルをツール・パレットで選択します。

アクティビティーの追加を停止したい場合には、**Esc** キーを押すか、またはツール・パレットで矢印をクリックします。

プロセス・ダイアグラムの保管

定義機能データベースにダイアグラムを保管するには、次のようにしてください。

1. 「プロセス」 ツリー・ビューで、保管したいプロセスを選択します。

2. プロセスを右クリックしてから、「**保管 (Save)**」をクリックします。

プロセス・プロパティーの変更を保管する方法についての詳細は、オンライン・ヘルプを参照してください。

プロセス・ダイアグラムの作図の指針

プロセス・ダイアグラムを作図するときには、ダイアグラム内でのアクティビティーの順序に注意してください。

1. 制御コネクターを作成して、アクティビティーの実行順序を示します。
2. アクティビティーからの出力データがあとのアクティビティーで入力として必要になる場合を示すデータ・コネクターを作成します。あるいは、出力データを、繰り返し実行されるアクティビティー (またはブロック) の入力データにマップしたい場合は、データ・コネクターを同じアクティビティー (またはブロック) から発してそこに戻るループとして作成します。また、入力データを出力データにマップすることもできます。
3. 2 つのアクティビティー間を結ぶ各データ・コネクターに、対応する制御コネクターがあるようにします。
4. より後のアクティビティーから前のアクティビティーに向かって制御またはデータ・コネクターを引かないでください。MQSeries Workflow は、ダイアグラム内でそのようなサイクルが偶然に作成されるのを防ぎます。
5. 繰り返し行う一連のアクティビティーがある場合は、それらのアクティビティーをブロック内に入れて、終了条件を指定してください。
6. ダイアグラムが大きいか、複雑である場合は、その外観を単純化して、複雑さのレベルを整理して示すことができるように、サブプロセスか、ブロックを使用することを考慮してください。他のプロセスでアクティビティーのセットを再使用したい場合には、サブプロセスにこれらのアクティビティーを定義することができます。終了条件に適合するまで繰り返されるアクティビティーのセットがある場合には、ブロックを使用できます。ブロックは do-until ループとして機能します。
7. 他のプロセス・アクティビティーを含むプロセスを開始するプロセス・アクティビティーを作図する場合は、プロセスの呼び出し順序を十分に調べてください。プロセスは、他のプロセスのインスタンスを任意の順序で開始したり、そのプロセス自体の他のインスタンスを開始することができます。



ダイアグラムを整理して表示するために、作図エリアで格子を使って、アクティビティーの位置を決めることができます。「**形式設定 (Format)**」メニューで、「**格子 (Grid)**」をクリックし、「**格子へのスナップ (Snap to Grid)**」を選択します。

シンボルを移動する方法についての詳細は、33ページの『プロセス・ダイアグラムでのオブジェクトの移動』を参照してください。

プロセス・ダイアグラム内のノードをコネクターと結合する

2つのアクティビティー間のコネクターを任意の組み合わせでダイアグラムに追加できます。コネクターの特性は、表2に示されています。

表2. プロセス内のコネクター



制御コネクターは、分岐条件に従って、プロセス内のアクティビティーの順序を指定します。



デフォルト・コネクターは、アクティビティーを終了している他の制御コネクターの分岐条件がどれも真として評価されないときに、アクティビティーの順序を指定します。



データ・コネクターは、あるアクティビティーから別のアクティビティーへのデータの流れを指定します。



データ・デフォルト・コネクターは、入力コンテナーから同一のアクティビティーの出力コンテナーへのデータの流れを指定します。



データ・ループ・コネクターは、出力コンテナーから同一のアクティビティーの入力コンテナーへ "戻る" データの流れを指定します。

注: ループ・コネクターを描画するときは、屈曲点を使用して、アクティビティーのまわりに、必ずループを作成してください。そうすることによって、ループ・コネクターの位置がアクティビティーに近くなりすぎることを防ぎます。ループ・コネクターの位置がアクティビティーに近くなりすぎると、再度それを選択してプロパティーを変更することができません。

詳細については、オンライン・ヘルプを参照してください。

以下の組み合わせでデータ・コネクターを追加することができます。

- ソース・ノードからアクティビティーへ
- アクティビティーからシンク・ノードへ
- 同一のアクティビティーの入力コンテナーから出力コンテナーへ
- 同一のアクティビティーの出力コンテナーから入力コンテナーへ
- 1つのアクティビティーの出力コンテナーから、後続のアクティビティーの入力コンテナーへ

コネクタを追加するには、ダイアグラム内にノードが 2 つ以上必要です。

アクティビティーを結合するには、以下の手順で行ってください。

1. 使用したいコンテナ・シンボルをクリックします。
2. マウス・ポインターをアクティビティー・ノード、ソース・ノード、またはシンク・ノードに移動し、コネクタを開始したい位置で 1 度クリックします。
3. マウス・ポインターをターゲット・ノードまで移動し、1 度クリックします。

これによって、シンボル間に線が引かれます。



屈曲点を作成したい地点で 1 度クリックすることによって、コネクタを描いたとおりに曲げることができます。屈曲点によって、コネクタを別の方向へつなげることができます。

また、屈曲点を削除し、これを後で既存のコネクタに追加することもできます。

サブプロセス用のデータ・コンテナの追加

表3 に示されているように、ソース・シンボルとシンク・シンボルは、入力データの受け渡しと、プロセス・アクティビティーまたはブロック・アクティビティーから出力データを収集するのに使用されるデータ・コンテナを表しています。

表3. プロセスまたはブロックのデータ・コンテナ



ソース・コンテナ (入力データ・コンテナ) には、サブプロセスまたはブロックへの入力として使用されるデータが入ります。



シンク・コンテナ (出力データ・コンテナ) には、サブプロセスまたはブロックからの出力として送られるデータが入ります。

入力データ・コンテナと出力データ・コンテナはそれぞれ 1 つしか使用できません。



複数のプログラム・アクティビティーのデータ・コンテナーは、1つのプロセス・ダイアグラム内のシンボルでは表されません。詳細については、39ページの『第4章 スタッフの割り当てとプロセス・フローの定義』を参照してください。

プロセス・アクティビティーとブロック・アクティビティーだけに、ソースおよびシンク・ノードがあります。

ソース・ノードまたはシンク・ノードをダイアグラムに追加するには、以下の手順で行ってください。

1. ツール・パレットから、ダイアグラムに含めたいソースまたはシンク・シンボルをクリックします。
2. マウス・ポインターを、ダイアグラム内でノードを置きたい位置まで移動し、1度クリックします。

ダイアグラムに描画したノードを移動する方法についての詳細は、33ページの『プロセス・ダイアグラムでのオブジェクトの移動』を参照してください。

プロセスのプロパティーの指定

新規のプロセスを作成するときには、プロセスのプロパティーが表示されます。「一般」ページが最初にオープンし、ここで、プロセスの名前その他の定義を指定します。たとえば、次のように指定できます。

プロセス開始時にデータをインプット

プロセスの入力コンテナー内にある未設定のデータ項目を初期設定するように、MQSeries Workflow がプロセス開始者にプロンプトを出すよう指定するには、この選択項目を選択します。

「データ」タブを選択すると、プロセスの入出力データ・コンテナーを記述するデータ構造を定義できます。データ構造オブジェクトをこれらのフィールドにドラッグ・アンド・ドロップして、「デフォルト・データ構造」項目を置き換えることができます。あるいは、「検出 (Find)」ボタンを使用して、データ構造の探索を行うことができます。



プロセス・ダイアグラムでアクティビティーの背景となる論理を指定する方法については、39ページの『第4章 スタッフの割り当てとプロセス・フローの定義』を参照してください。また、データ構造の詳細については、35ページの『データ構造の定義』に説明されています。

フィールドに何を入力するかについての詳細は、オンライン・ヘルプを参照してください。

プロセス・ダイアグラムでのオブジェクトの移動

プロセス・ダイアグラムでは、ノード、屈曲点、またノードやコネクターのテキスト・フィールドを移動できます。

ノード、屈曲点、またはテキスト・フィールドを移動するには、以下の手順で行ってください。

1. マウス・ポインターで、移動したいノードをクリックします。
2. 移動先までノードをドラッグします。
3. ノードの位置が決まったら、マウス・ボタンを放し、ノードをドロップします。

ノードをプロセス・ダイアグラム内の別の位置に移動すると、それらのノードに接続されているコネクターは、移動先の位置まで伸びます (あるいは縮みます)。格子へのスナップを設定すると、移動した各ノードと、各コネクターの屈曲点は、1 つの格子の交点に中央がそろいます。



ノードの移動をやめる場合は、以下のようにします。

- ドラッグ・アンド・ドロップ操作中のマウス・ボタンを放していないなら、**Esc** キーを押します。移動したオブジェクトは、元の位置に戻ります。
- ドラッグ・アンド・ドロップ操作を終了している場合は、変更を保存せずにダイアグラムを終了して元に戻すしかありません。

プロセス・ダイアグラムの部分コピー

同じプロセス・ダイアグラム内で、またはあるプロセス・ダイアグラムから別のプロセス・ダイアグラムへ、プロセスを部分的にコピーして貼り付けることができます。コネクターをコピーする場合は、コピーするセグメントに起点ノードとターゲット・ノードを組み込みます。

コピーして貼り付けるには、以下の手順で行います。

1. コピーしたいダイアグラムのセグメントを選択します。

ダイアグラムの大きなセグメントを選択するには、左マウス・ボタンをクリックして、コピーしたい範囲を長方形で囲みます。左マウス・ボタンを放すと、長方形の中のすべてのノードとその間のコネクターが選択されます。

あるいは別の方法として、Ctrl キーを押したまま、左マウス・ボタンをクリックして、ノードとコネクターを幾つか選択することもできます。Ctrl キーを押したままにする必要があるのは、マウス・ボタンをクリックする間だけです。

2. メニュー・バーの「**編集 (Edit)**」をクリックしてから、「**コピー (Copy)**」をクリックします。
プロセスから選択した部分はクリップボードにコピーされます。
3. プロセス・セグメントを挿入したいダイアグラムを活動化します。
4. メニュー・バーの「**編集 (Edit)**」をクリックしてから「**貼り付け (Paste)**」をクリックして、コピーしたプロセス・セグメントをクリップボードからダイアグラムに挿入します。
5. セグメントをドラッグして、ダイアグラム内の配置したい場所に移動します。
プロセス・セグメントの貼り付けをやめる場合は、**Esc** キーを押します。
プロセスはダイアグラムに貼り付けられません。
6. コピーしたプロセス・セグメントをクリックして、その場所に固定します。

プロセス・セグメントをクリップボードにコピーすると、そこではプロセス・セグメントが FDL 形式になります (73ページの『第6章 FDL ファイル内でのワークフロー情報の定義』を参照)。このテキストをテキスト・エディターに貼り付けて、定義を変更することができます。その後、変更した FDL 定義をクリップボードに戻し、変更したプロセス・セグメントとしてプロセス・ダイアグラムに貼り付けます。

注:

1. FDL ファイルを変更する場合、米国規格協会 (ANSI) コード・ページを使用しているテキスト・エディターを選択して、コード・ページが異なることによって矛盾が生じることのないようにしてください。
2. FDL ファイルをクリップボードに追加する場合、ファイルに FDL ヘッダーが含まれていることを確認してください。

プロセスのセグメントを切り取ってプロセス・ダイアグラム内で新しい位置に移動したい場合は、メニュー・バーの「**編集 (Edit)**」をクリックしてから、「**切り取り (Cut)**」をクリックします。コピーして貼り付ける手順の「**コピー (Copy)**」を「**切り取り (Cut)**」に読み替えてください。

プロセス・ダイアグラムの部分削除

プロセス・ダイアグラムでは、ノード、コネクタ、または屈曲点を削除できます。

削除は以下の手順で行います。

1. 削除したいオブジェクトを右クリックします。
2. 「**削除**」をクリックするか、または **Delete** (削除) キーを押します。

1 つの項目を選択するには、その項目でクリックします。個々の項目が集まったグループを選択する場合は、Ctrl キーを押しながらそれらの項目グループをクリックします。詳細については、オンライン・ヘルプを参照してください。

データ構造の定義

MQSeries Workflow では、データ構造定義は、プロセス、アクティビティ、およびブロックの入力および出力データ・コンテナの内容を記述します。入力または出力として使用されるデータ、あるいは終了条件または分岐条件で参照されるデータは、データ構造定義で説明する必要があります。

各データ構造は、いくつかのメンバーによって構成されます。たとえば、住所を定義するのに使用されるデータ構造には、市町村名および地区名のメンバーが入ります。

データ構造メンバーのデータ・タイプは、基本的な MQSeries Workflow のデータ・タイプ (ストリング形式、長形式、浮動小数点形式、2 進形式) のいずれかにすることもできますし、以前に定義された別のデータ構造を参照することもできます。別のデータ構造を参照するデータ構造をネストされた データ構造といいます。

データ構造 A にデータ構造 B がメンバー・タイプとして含まれている場合、あるいはデータ構造 B にデータ構造 A がメンバー・タイプとして含まれている場合、以下の例に示されているように正常に機能しません。

```
STRUCTURE 'A'  
  'Member': 'B';  
END 'A'  
STRUCTURE 'B'  
  'Member': 'A';  
END 'B'
```

プログラム登録や、プロセスおよびアクティビティ定義で参照する前に、データ構造を定義しなければなりません。



コンテナおよびデータ構造についての詳細は、*IBM MQSeries Workflow*: プログラミングの手引き を参照してください。

デフォルト・データ構造

MQSeries Workflow は、1 つのデータ構造オブジェクト、デフォルト・データ構造 を事前に定義しています。ツリー・ビューで最初に「インプリメンテーション」タブをクリックすると、すでに作成されたこのオブジェクトが「データ

構造」のもとに表示されます。このデフォルト・データ構造を削除または名前変更することはできません。必要であれば、ユーザーが定義したメンバーを、デフォルト・データ構造に追加することができます。



各プログラム、プロセス、およびブロック・アクティビティーのプロパティーには、入力および出力データ構造のデフォルト設定として、デフォルト・データ構造が含まれます。

自分で作成した他のデータ構造を参照するときは、デフォルト設定を変更することができます。デフォルト・データ構造を変更するときは、データ構造の割り当てられるプロセス・モデルも変更されます。

データ構造の定義

新規のデータ構造を定義するには、次の手順で行います。

1. データ構造を作成します。
2. データ構造のプロパティーを指定します。
3. データ構造のメンバーを定義します。

データ構造を作成するには、次のようにします。

1. 「インプリメンテーション」ツリー・ビューで、「データ構造」を右クリックします。
2. 「新規のデータ構造」をクリックします。
「データ構造のプロパティー」ウィンドウが表示されます。
3. データ構造の名前を含む設定を入力します。

フィールドに何を入力するかについての詳細は、オンライン・ヘルプを参照してください。

プログラムの登録

モデルのプログラム・アクティビティーは、実行時にこれらのプログラムにアクセスできなければなりません。そのためには、MQSeries Workflow にプログラムを登録しなければなりません。使用する実行可能プログラムの名前を指定しなければなりません。さらに、以下の情報を指定することもできます。

- プログラムの実行環境
- プログラムによって使用される MQSeries Workflow の入力データ構造
- プログラムによって使用される MQSeries Workflow の出力データ構造
- 実行時にプログラムに渡されるパラメーター

プロセス内のプログラム・アクティビティーは、プログラムの実際の名前ではなく、プログラム登録名を参照するので、モデルは柔軟になります。登録において、プログラム、環境の特性、およびプログラムに渡されるパラメーターを変更することができます。

プログラム登録に関連付けられているプログラムを変更する場合は、そのプログラム登録を使用するプロセスを保管し直す必要はありません。実行機能にワークフロー・モデルを搬入すると (49ページの『第5章 ワークフロー・モデルの操作可能なプロセスへの変換』を参照)、新しい定義が有効になります。しかし、実行機能にワークフロー・モデルをすでに搬入している場合には、これを定義機能から搬出し、これを実行機能に搬入し直して、新しい定義を使用できるようにします。

MQSeries Workflow とともに実行するアプリケーションの設計、およびアプリケーション・プログラム・インターフェース (API) の使用方法については、*IBM MQSeries Workflow: プログラミングの手引き* を参照してください。



プログラムは、実行機能で開始されるそれぞれのオペレーティング・システムに定義されていなければなりません。

プログラム登録オブジェクトを作成するには、次のようにします。

1. 「インプリメンテーション」 ツリー・ビューで、「プログラム」を右クリックします。
2. 「新規のプログラム」をクリックします。
「プログラムのプロパティー」ウィンドウが表示されます。
3. これらの設定値を入力した後で、「了解」をクリックしてプログラム登録を確認します。

フィールドに何を入力するかについての詳細は、オンライン・ヘルプを参照してください。

第4章 スタッフの割り当てとプロセス・フローの定義

この章では、スタッフをアクティビティーに割り当てる方法、およびプロセス・フローを定義する方法について説明します。これらの定義は、プロセスを実行時に開始するときに常に適用されます。

プロセス・ダイアグラム内のアクティビティーおよびコネクターの背景となる論理を定義するには、適切なプロパティー・ウィンドウを使用します。

たとえば、プログラム・アクティビティーのプロパティーをオープンするには、次のようにします。

1. 論理を定義したいプロセスのプロセス・ダイアグラムをオープンします。
2. プロパティーを定義したいプログラム・アクティビティーを右クリックします。
3. 「プロパティー」をクリックします。

「プログラム・アクティビティーのプロパティー」ウィンドウが表示されません。

アクティビティーのプロパティーの指定

プロセス・ダイアグラムに追加するアクティビティーそれぞれにつき、プロパティーを指定しなければなりません。これらのプロパティーは、実行時にプロセス・フローを判別するのに使用されます。

フィールドに何を入力するかについての詳細は、オンライン・ヘルプを参照してください。

アクティビティーへのスタッフの割り当て

実行機能のユーザーがプロセス・インスタンスを開始するとき、このプロセスの各アクティビティーには、1 つまたは複数のスタッフ・メンバーが割り当てられていなければなりません。

スタッフ・メンバーには次の 2 種類があります。

動的 動的スタッフ割り当てでは、割り当てられたアクティビティーをワーク・リストで受け取る人についてここに指定した基準を、MQSeries Workflow が実行時に解決します。アクティビティーが作動可能になっ

たときに、指定の基準を満たす MQSeries Workflow クライアントのエンド・ユーザーがそのアクティビティーを受け取ります。

指定する基準は、アクティビティーを受け取る人のレベル、組織、または役割と関連付けます。あるいはこれらの基準を組み合わせたものにすることができます。また、後ほど実行機能で解決されるコンテナ・メンバーをベースとすることもできます。

プロセス・インスタンス内の先行するアクティビティーの開始者に関する情報に基づいて、アクティビティーを人に割り当てることもできます。

動的割り当ての利点は、ユーザーのワークフロー・モデルにおいて割り当てに柔軟性をもたせることができる点です。スタッフに変更があった場合にモデルを変更する必要がありません。

特定 特定スタッフ割り当てでは、ワーク・リストで目的のアクティビティーを受け取る人のユーザー ID を指定します。これらの人しかアクティビティーを受け取ることができません。

特定ユーザーへのアクティビティーの割り当ては、動的スタッフ割り当てほど柔軟なものではありません。割り当てられた人がその所属先で仕事を変えたり、仕事そのものをやめてしまった場合、その割り当ては無効になります。その担当者オブジェクトに他に適当な人を割り当てていないかぎり、ワークフロー・モデルを変更しなければなりません。

しかし、プロセスのテストを行っている場合、あるいは、あるアクティビティーを実行できる人が特定の決まった人のみである場合は、特定スタッフ割り当てが役立ちます。

特定スタッフ割り当てを使い、その割り当てが解決できないとき、MQSeries Workflow はこのアクティビティーに対する割り当てのタイプを動的割り当てに変更します。その後、MQSeries Workflow はその割り当てを解決しようとします。この場合、「スタッフ」ページでプロセスに指定されるプロパティー (役割、組織)、および「**スタッフ 2**」ページのアクティビティーに指定されるプロパティー (役割、組織、レベル範囲) が使用されます。

動的スタッフ割り当ての指定

「**スタッフ 2**」ページに、アクティビティーを開始する人が満たしていなければならない基準を入力します。41ページの図8は、「**スタッフ 2**」ページを示しています。ここに、これらの基準を入力します。

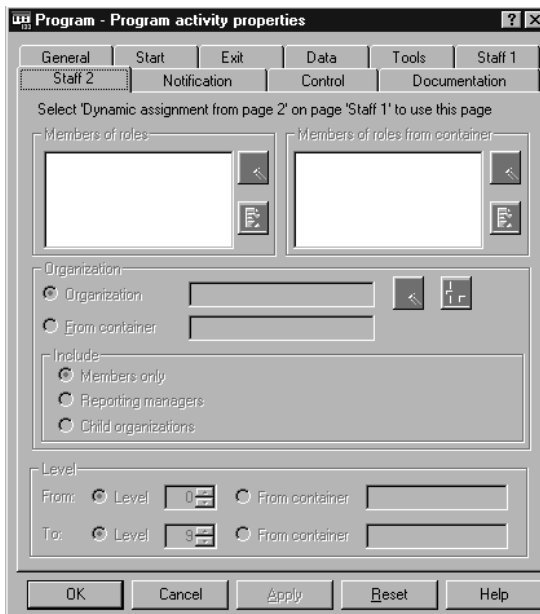


図8. 「スタッフ 2」 ページ

「スタッフ 2」 ページの基準を使用するためには、「スタッフ 1」 ページで、「ページ 2 からの動的割り当て (**Dynamic assignment from page 2**)」を選択しなければなりません。42ページの図9 には、選択できる定義が示されています。

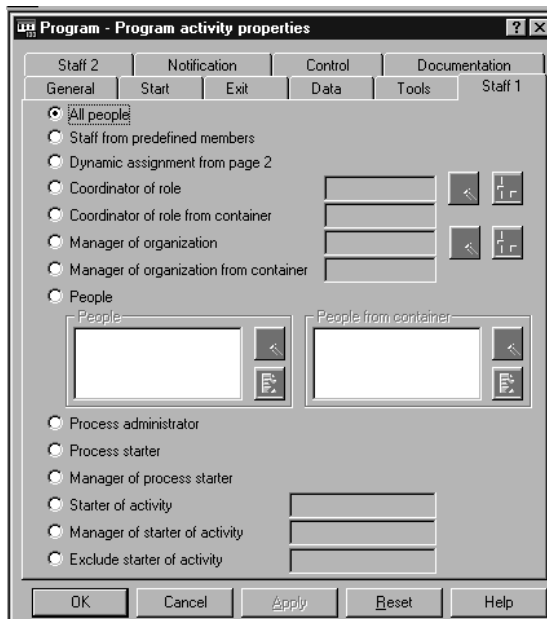


図9. 「スタッフ 1」 ページ

1 つのプロセス定義だけに基づいて動的スタッフ割り当てを行いたい場合は、動的割り当てを選択しますが、「スタッフ 2」ページの基準は何も選択しないでください。プロセスのプロパティは、32ページの『プロセスのプロパティの指定』およびオンライン・ヘルプで説明されています。



プロセスのインスタンスが開始し、アクティビティーが作動可能な状態になると、MQSeries Workflow は、動的割り当ての基準を使用して、そのアクティビティーの開始者として可能性のある人を識別します。この基準を満たす、すべての人がワーク・リストでそのアクティビティーを受け取ります。

その基準を満たす人がいないか、その基準を満たすすべての人が不在の場合は、プロセス管理者がそのアクティビティーを受け取ります。プロセス管理者は、他の人のアクティビティーにアクセスすることが許可されている場合、そのアクティビティーを転送することができます。

基準を指定しない場合、つまり、選択項目「ページ 2 からの動的割り当て (Dynamic assignment from page 2)」を活動化したものの、基準を選択しなかった場合、実行時に次のことが起こります。アクティビティーは、プロセス定義で指定されたものと同じ、役割および組織のユーザーのワーク・リストに表示されます。プロセス定義に基準が何も指定されていないと、プロセスを開

始する担当者のワーク・リストに、そのアクティビティーが表示されます。さらに、プロセス開始者と同じ組織に所属するすべての人のワーク・リストにもそのアクティビティーが表示されます。

アクティビティーにスタッフを割り当てるときに考慮する必要がある事柄を確認して、スタッフの割り当てが実行時に正常に行われるようにしたい場合は、123ページの『付録A. スタッフ定義のモデル化についての詳細』および51ページの『ワークフロー・モデルの検証』を参照してください。

コネクターの論理の定義

コネクターの背景となる論理を定義するには、**ダイアグラム・ビュー**をオープンし、プロセス・ダイアグラム内のコネクターをダブルクリックしてください。制御コネクターの場合、コネクターを記述する際のプロパティがこの操作でオープンします。データ・コネクターの場合、データ・コネクターを定義する際のプロパティがこの操作でオープンします。

プロセス・フローを制御するコネクター

制御コネクターは、アクティビティー間の流れを決定します。

「**一般**」ページでは、制御コネクターの名前と説明を指定できます。さらに、アクティビティーの分岐条件も指定できます。

分岐 これにより、ワークフローで使用する論理式を定義します。指定した条件が実行時に真の場合、制御は、その制御コネクターのターゲットに渡されます。オンライン・ヘルプに記載されている構文規則に従って、条件を記述する論理式を入力してください。

「**分岐**」ページを空のままにすると、分岐条件は真として評価され、制御の流れはこの制御コネクターに従います。



分岐条件で出力コンテナ変数の名前を使用し、アクティビティーまたはブロック名を指定しないと、デフォルトによって、制御コネクターの起点となるアクティビティーが使用されます。分岐条件でアクティビティーまたはブロック名を指定し、あとでその名前を変更する場合は、ここでもそれを更新しなければなりません。さらに、アクティビティー名を指定する場合は、参照されたアクティビティーから現行のアクティビティーへの制御パスがなければなりません。

データ・フローを制御するコネクタ

データ・コネクタは、起点アクティビティまたはブロックからターゲット・アクティビティまたはブロックへの流れを決定します。起点データ構造とターゲット・データ構造が同じで、ターゲット・アクティビティに結び付いているデータ・コネクタが他にない場合、MQSeries Workflow は、起点データ・コンテナからのデータを自動的にターゲット・データ・コンテナにマップします。



ユーザー定義のメンバーは、データ・コンテナの詳細ビューで、`_STRUCT` 項目の下に表示されます。

2 つのコンテナのデータ構造が異なる場合、あるいは、ターゲット・アクティビティへ結合されている別のデータ・コネクタがある場合は、データの流れをマップする必要があります。事前定義されたデータ構造メンバーを使用したい場合には、これらもマップしなければなりません。異なるソースから単一のデータ項目にデータをマップすることもできます。

データ構造についての詳細は、*IBM MQSeries Workflow: プログラミングの手引き* を参照してください。

データ・コンテナ間でのデータのマッピング

起点アクティビティまたはブロックからターゲット・アクティビティまたはブロックへマップを開始するには、プロセス・ツリー・ビューで起点アクティビティを右クリックします。



別の方法として、プロセス・ダイアグラムで直接作業することもできます。

- プロセス・ダイアグラムで、アクティビティまたはデータ・コネクタを右クリックします。
- ショートカット・メニューで「**コンテナ・マッピング**」をクリックします。

ソースおよびシンク・シンボルのマッピングに関する情報の詳細は、オンライン・ヘルプを参照してください。

このようにして、46ページの図10 に示されているように、「**コンテナ・マッピング、マッピング先、マッピング元**」を選択します。

マッピング先

ウィンドウの左の区画では、AssessRisk などの選択されたアクティビ

ティ어의出力コンテナが表示されます。右の区画では、AcceptCredit および RequestApproval などの、ターゲット・アクティビティの入力コンテナが表示されます。

マッピング元

ウィンドウの右の区画には選択されたアクティビティの入力コンテナが表示され、左の区画では、すべてのソース・アクティビティの出力コンテナが表示されます。

アクティビティの「中に」マップしたい場合には、30ページの『プロセス・ダイアグラム内のノードをコネクタと結合する』で説明されているように、データ・ループ・コンテナまたはデータ・デフォルト・コンテナを使用することができます。

コンテナ・マッピング・ダイアログ・ボックスが現れると、起点および宛先アクティビティの両方のデータ・コンテナが、1つのウィンドウに表示されます。

このウィンドウは縦に分割され、2つの画面区画に分かれています。左の区画には起点コンテナが表示され、右の区画にはターゲット・コンテナが表示されます。両方の部分はそれぞれ別個にスクロールできます。起点アクティビティが複数あったり、選択できるターゲットが複数あったりする場合には、コンテナ・メンバーは重ねてリストされます。

起点 これは、起点アクティビティまたはブロックのためのプロパティの「データ」ページに指定されている出力データ構造と、事前定義データ構造メンバーを示しています。データ・コネクタの起点がプロセスまたはブロックのソース・ノードである場合、そのプロセスまたはブロックの入力データ構造が表示されます。

プロセスまたはブロックの入力コンテナからターゲットの入力コンテナにマップします。

ターゲット

これは、ターゲット用のプロパティの「データ」ページに指定されている入力データ構造と、事前定義データ構造メンバーを示しています。データ・コネクタのターゲットがプロセスまたはブロックのシンク・ノードの場合には、そのプロセスまたはブロックの出力データ構造が表示されます。起点出力コンテナからプロセスまたはブロックの出力データ・コンテナへマップします。

データ・コンテナ・メンバーのタイプはすべて同じでなくてはなりません。たとえば、ストリング・タイプのメンバーは、浮動タイプのメンバーへはマップすることができません。

両方のメンバー項目が正確に一致すれば、複合メンバー、つまり、(ユーザー定義) `_STRUCT`、ネストされたデータ構造、または配列を別の複合メンバーにマップすることができます。ネストしたデータ構造および `_STRUCT` は、ターゲットが起点と同じ名前でないとはマップできません。名前が異なる複合データ構造を互いに直接マップするには、メンバーごとにそれらをマップしてください。

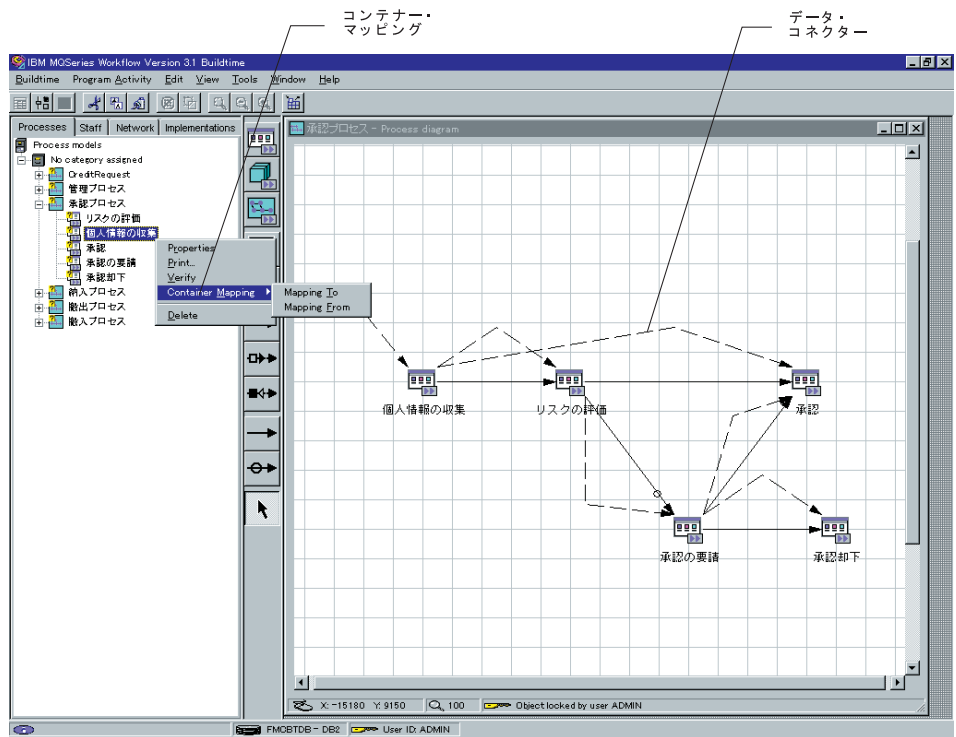


図 10. データ・フローの定義

図10 は、アクティビティー間のデータ・コネクターのダイアグラム・ビューと、ツリー・ビューの「コンテナ・マッピング」メニューを示しています。

ドラッグ・アンド・ドロップによるマッピング: ドラッグ・アンド・ドロップでデータ構造メンバーをマップするには、次のようにします。

1. 起点データ・コンテナの詳細ビューから、データ構造メンバーをドラッグします。
2. それをターゲット・データ・コンテナの詳細ビュー内のデータ構造メンバー上にドロップします。



両方のメンバーが正確に一致すれば、複合メンバー、(つまり、`_STRUCT`、ネストされたデータ構造、または配列) を別の複合メンバー上にドロップすることができます。ネストしたデータ構造および `_STRUCT` は、ターゲットが起点と同じ名前でないとはマップできません。 `_STRUCT` は、ユーザーが定義するデータ構造全体を表しますが、事前定義されたデータ構造メンバーはここに含まれません。

事前定義データ構造メンバーのマッピング

IBM MQSeries Workflow: プログラミングの手引き に示されているように、事前定義データ構造メンバー `_PROCESS_INFO` および `_ACTIVITY_INFO` を使用する場合は、以下のメンバーを起点データ・コンテナ内のメンバーからターゲット・データ・コンテナ内のメンバーに明示的にマップしなければなりません。

固定の事前定義データ構造のメンバー (`_RC`、`_PROCESS` および `_ACTIVITY`) もマップする必要がありますが、これらは起点出力コンテナでのみ表示されます。これらをマップするには、ターゲット・データ構造内に、マップ先としてのデータ構造メンバーを定義しておく必要があります。

データ・コンテナ・メンバーのデフォルトの指定

データ・コンテナ・メンバーのメンバー項目を初期設定するためのデフォルト値を指定することができます。配列要素については、このデフォルト値は単純データ・タイプでなければなりません。ネストされたデータ構造については、デフォルト値を指定することはできません。

「マッピング」ダイアログで、データ・コンテナ・メンバーのデフォルト値を指定するには、直接フィールドを編集できます。

詳細については、オンライン・ヘルプを参照してください。

第5章 ワークフロー・モデルの操作可能なプロセスへの変換

この章では、実行機能で使用するプロセス・モデルを検証する方法について説明します。この章では、完成したワークフロー・モデルを定義機能から搬出し、これを MQSeries Workflow 実行機能に搬入する方法についても説明します。これらのステップは、プロセス・テンプレートを作成するのに必要であり、実行機能の許可ユーザーは、ここから実行可能プロセス・インスタンスを作成することができます。

8ページの『定義機能と実行機能が協働する方法』に説明されているとおり、定義機能から実行機能へのワークフロー・モデルの自動転送はなく、その逆もありません。ユーザー定義アイコンの自動転送もありません。自分で作成したアイコンをツール・パレットに使用したい場合には、*IBM MQSeries Workflow: インストールの手引き* で説明されている方法でこれらのアイコンをインストールしなければなりません。ユーザー定義のアイコンは定義機能と実行機能の両方で利用可能にし、これらのアイコン用には必ず適切なインストール・パスを使用してください。

定義機能にはワークフロー・モデル情報を搬出および搬入するための組み込み機能がありますが、実行機能は、サーバー・インストールの一部であるコマンド行インターフェースを使用します。

さらにこの章では、MQSeries Workflow バージョン 3.1 または FlowMark (R) バージョン 2.3 で作成したワークフロー・モデルを使用する方法についても説明します。

定義機能および実行機能でのワークフロー定義の使用

メイン・データベースとされている実行機能データベースがプロセスの実行に使用されます。プロセスを定義機能データベースで定義するか、または MQSeries Workflow の外で FDL ファイルを作成します。FDL ファイルを定義機能に搬入する際に、FDL ファイルが MQSeries Workflow 実行機能から発生したものか、MQSeries Workflow 以外から得たものかを指定できます。

実行機能用のオブジェクトの状況の定義

データベースの同期を維持するために、定義機能はオブジェクト状況を示すフラグを使用します。データベースを同期化する方法についての詳細は、8ページの『定義機能と実行機能が協働する方法』を参照してください。

実行機能に必要な定義を変更したい場合には、これらの変更を定義機能で実行します。これは、新しい定義を作成する場合にも適用されます。定義機能のツリー・ビューでは、すべての主なオブジェクトにそれらの状況を示すフラグが付けられます。状況シンボルは、ツリー・ビューでオブジェクトの左側に付きます。

オブジェクト状況シンボルのリストについては、オンライン・ヘルプを参照してください。

以下のことに留意してください。

- 実行機能から搬出する FDL ファイルでは、この出所が実行機能データベースであることを、オブジェクトの状況が示します。FDL ファイルを定義機能に搬入すると、これらのオブジェクトは実行機能と定義機能データベースの両方に存在します。
- FDL ファイルが MQSeries Workflow 実行機能の外で作成されてそれから定義機能に搬入されると、オブジェクトの状況は以下のようになります。

更新 (Updated)

これが定義機能に存在しているオブジェクトの場合には、同じものが実行機能にも存在することになり、したがって、更新とみなされます。

不確定 (In Question)

オブジェクトが実行機能に存在すると示されていないため、定義機能で新規とみなされます。

定義機能でオブジェクト状況を変更したい場合には、以下のことを実行できません。

すべてのオブジェクトをデフォルトにリセットするには、次のようにします。

1. メニュー・バーで「**定義機能**」をクリックします。
2. 「**オブジェクトの状況**」をクリックします。
3. 「**すべてをデフォルトにリセットする**」を選択します。

個々のプロセスを**更新**に設定するには、次のようにします。

1. メニュー・バーで「**オブジェクトの状況**」をクリックします。
2. 「**更新にリセットする**」を選択します。

詳細については、オンライン・ヘルプを参照してください。

ワークフロー・モデルの検証

定義機能で定義したワークフロー・モデルや、FDL ファイルの中にあるワークフロー・モデルを、実行機能での使用に備えて変換する前に、それらのモデルに誤りがないかをチェックします。実行機能搬入ユーティリティーで変換オプションを使用すれば、それらのチェックがモデルの変換中に自動的に実行されます。変換オプションを使用する方法についての詳細は、66ページの『搬出 / 搬入ユーティリティーのオプション』を参照してください。

定義機能のプロセス・ダイアグラムの作業中に、モデルの検証を行うこともできます。次のように行います。

1. メニュー・バーで「プロセス」をクリックします。
2. 「検証 (Verify)」をクリックします。

これにより、『ワークフロー・モデルの検証に関する規則』に示されている数多くのチェックが開始されます。

ワークフロー・モデルの検証に関する規則

ワークフロー・モデルを検証または変更するときには、次のように様々なチェックが実行されます。

- プロセスについてのチェック
- プロセスとプロセスのアクティビティーについてのチェック
- すべてのアクティビティーについてのチェック
- プログラム・アクティビティーとプロセス・アクティビティーについてのチェック
- プロセス・アクティビティーについてだけのチェック
- ブロック・アクティビティーについてのチェック
- プログラム・アクティビティーについてだけのチェック
- 制御コネクターについてのチェック
- データ・コネクターについてのチェック
- データ構造についてのチェック

プロセス・モデルが実行時に正常に使用されるためには、そのモデルが以下に示す規則に従っている必要があります。

プロセスについてのチェック

- ダイアグラムを空にすることはできません。ダイアグラムの中には最低でも 1 つのアクティビティーを含める必要があります。

- 「**コンテナにあるプロセス期間 (Duration of process From container)**」を定義する場合には、長形式タイプのデータ構造メンバーが存在していなければなりません。詳細については、35ページの『データ構造の定義』およびオンライン・ヘルプを参照してください。
- 以下の定義のいずれかを使用する場合には、ストリング・タイプのデータ構造メンバーが存在していなければなりません。
 - 「**コンテナにあるプロセス管理者 (Process Administrator From container)**」
 - 「**コンテナにある組織 (Organization From container)**」
 - 「**コンテナにある役割 (Role From container)**」

これらの定義を作成または変更するには、「**プロセス・プロパティー (Process properties)**」ウィンドウをオープンし、「**スタッフ**」タブをクリックします。プロパティーを定義する方法についての詳細は、25ページの『プロセスの作成とそのプロパティーの指定』、32ページの『プロセスのプロパティーの指定』、およびオンライン・ヘルプを参照してください。

プロセスとプロセスのアクティビティーについてのチェック



データ構造を定義機能で定義する場合、それらのデータ構造は実行時にデータ・コンテナのテンプレートとしての役目を果たします。つまり、アクティビティーについて、「**コンテナにある (From container)**」を指定すると、入力コンテナに保管されたデータが実行時にアクティビティーやプロセスのために使用されることとなります。

以下の規則が、プロセス、プログラム・アクティビティー、プロセス・アクティビティー、およびブロック・アクティビティーに適用されます。

- 入力データ構造と出力データ構造が存在していなければなりません。データ構造を定義する方法についての詳細は、35ページの『データ構造の定義』およびオンライン・ヘルプを参照してください。
- 入力コンテナと出力コンテナの両方の初期値の数は次のように限定されています。各コンテナの初期値の内部表示は 32 KB を超えてはならない。

注: コンテナ値は実行時に設定されます。

- 入力コンテナと出力コンテナのデフォルト値については、以下の規則が適用されます。

- 読み取り専用の事前定義入力コンテナまたは出力コンテナのメンバーに、デフォルト値 `_PROCESS`、`_PROCESS_MODEL`、`_ACTIVITY`、および `_RC` を含めてはなりません。 `_RC` は唯一の出力コンテナ・メンバーであることに注意してください。
- デフォルト値を設定する入力コンテナまたは出力コンテナのメンバーは、関連するデータ構造に存在していなければなりません。これには、非配列のメンバーを配列としてアドレス指定することや、配列のメンバーを非配列としてアドレス指定することが含まれます。
- デフォルトのメンバーを設定する入力コンテナまたは出力コンテナのメンバーは、基本タイプが含まれていなければなりません。つまり、そのようなメンバーが副構造や配列であってはなりません。
- 入力コンテナまたは出力コンテナのメンバーのデフォルト値は、メンバー・タイプの構文規則に従っていなければなりません。たとえば、ストリング `abc` を長形式のメンバーに割り当てることはできません。

定義を変更するには、データ・コネクタを右クリックします。ショートカット・メニューで、「マッピング」をクリックします。「コンテナ・マッピング (container mapping)」ダイアログがオープンします。詳細については、44ページの『データ・コンテナ間でのデータのマッピング』を参照してください。

すべてのアクティビティについてのチェック

- 注:** プロセス内にあるアクティビティとアクティビティの間の流れを決定するためには、制御コネクタを使用します。あるアクティビティから他の後続のアクティビティへの接続のことを制御パスと言います。
- 終了条件は有効なブール式でなければなりません。有効なブール式については、81ページの『条件の構文』で定義されている条件構文を参照してください。終了条件として使用するデータ構造のメンバーはすべて、出力データ構造の中に存在していなければなりません。また、それらのメンバーの中には、使用されるコンテキストに応じて適切なタイプが含まれていなければなりません。他のアクティビティからのデータ構造メンバーを使用する場合は、そのアクティビティから現在チェックしているアクティビティまでの制御パスが存在していなければなりません。
 - 入ってくる制御コネクタを 255 以上指定してはなりません。

- 不完全な分岐条件が指定された出ていく制御コネクタを指定し、かつ出ていくデフォルト・コネクタが少なくとも 1 つは存在している場合、警告が送られてきます。その警告は、不完全な分岐条件が TRUE として評価されるために、出ていくデフォルト・コネクタが使用できないことを通知します。
- アクティビティ記述の置換変数として使用されるデータ構造メンバーがすべて存在していなければなりません。

プログラム・アクティビティとプロセス・アクティビティについてのチェック 以下に示す一般的な規則が、プログラム・アクティビティとプロセス・アクティビティに適用されます。

- **サポート・ツール**として定義したプログラムはすべて存在していなければなりません。
- 最低でも 1 つのプラットフォームのプログラム・プロパティを定義する必要があります。使用できるプラットフォームは、Windows NT、Windows 9x、OS/2、AIX、HP-UX、または OS/390 です。
- 関連プログラムに OS/390 外部サービスが含まれている場合には、以下のプログラム・プロパティを設定する必要があります。
 - 「サービス (Service)」
 - 「呼び出しタイプ (Invocation type)」
 - 「実行可能プログラム (Executable)」
 - 「実行可能プログラム・タイプ (Executable type)」
- 定義可能なサポート・ツールの数は限定されています。次の式が適用されます。サポート・ツールの名前の長さ (バイト数) を、実際に使用したいサポート・ツールの数に追加する。その合計が 254 バイトを超えてはならない。
- プログラム・アクティビティに適用されるチェックはサポート・ツールにも適用されます。
- 以下の定義のいずれかを使用する場合には、長形式タイプのデータ構造メンバーが存在していなければなりません。
 - 「コンテナにある優先順位 (Priority From container)」
 - 「コンテナにあるアクティビティ期間 (Duration of activity From container)」
 - 「コンテナにある決定期間 (Duration of making decision From container)」
- 以下の定義のいずれかを使用する場合には、ストリング・タイプのデータ構造メンバーが存在していなければなりません。

- 「コンテナーにある遅延通知人 (Person to notify of delay From container)」
- 「コンテナーにある組織マネージャー (Manager of organization from container)」
- 動的スタッフ割り当てを使用する場合は、以下の点もチェックされます。
 - 「下限レベル (From level)」を定義する場合は、0～9 の値を必ず選択しなければなりません。ただし、「下限レベル (From level)」と一緒に「コンテナーにある (From container)」も定義する場合は例外です。この場合には、長形式タイプのデータ構造メンバーが存在していなければなりません。
 - 「上限レベル (To level)」を定義する場合は、0～9 の値を必ず選択しなければなりません。ただし、「上限レベル (To level)」と一緒に「コンテナーにある (From container)」も定義する場合は例外です。この場合には、長形式タイプのデータ構造メンバーが存在していなければなりません。
 - 「下限レベル (From level)」 / 「コンテナーにある (From container)」も、「上限レベル (To level)」 / 「コンテナーにある (From container)」も定義しない場合は、「下限レベル (From level)」の値を「上限レベル (To level)」で指定した値以下にしなければなりません。
 - 以下の定義のいずれかを使用する場合には、ストリング・タイプのデータ構造メンバーが存在していなければなりません。
 - 「コンテナーにある役割メンバー (Members of roles From container)」
 - 「コンテナーにある組織 (Organization From container)」
- 優先順位を入力コンテナーから取り出さない場合、優先順位は 0～999 の範囲の数値でなければなりません。
- 以下の定義のいずれかを使用する場合には、ストリング・タイプのデータ構造メンバーが存在していなければなりません。
 - 「コンテナーにある人 (People from container)」
 - 「コンテナーにある役割コーディネーター (Coordinator of role from container)」
- 以下の定義のいずれかを使用する場合には、選択を行うアクティビティ (たとえば、「アクティビティの開始者 (Starter of activity)」) が存在していなければなりません。さらに、そのアクテ

イビティから現在チェックしているアクティビティまでの制御パスも存在していなければなりません。

- 「**アクティビティの開始者 (Starter of activity)**」
- 「**アクティビティの開始者のマネージャー (Manager of starter of activity)**」
- 「**アクティビティの開始者の除外 (Exclude starter of activity)**」

この選択を行う場合には、アクティビティのスタッフ割り当てに最低でも 2 人の人員を定義しなければなりません。

アクティビティに 1 人しか定義されていない場合、この人が行えるのはアクティビティを開始することだけです。アクティビティの開始者を除外すると、関連したスタッフ割り当てを行える人はなくなってしまいます。詳細については、124ページの『**アクティビティのプロパティを基にしたスタッフ解決**』を参照してください。

プログラム・アクティビティやプロセス・アクティビティのプロパティを作成または変更するには、「**プログラムまたはプロセス・アクティビティのプロパティ (Program or Process activity properties)**」ウィンドウをオープンします。「**制御 (Control)**」タブを選択して優先順位を定義するか、「**通知 (Notification)**」タブを選択して期間パラメーターを指定します。アクティビティを定義する方法についての詳細は、26ページの『**プロセス・ダイアグラムへのアクティビティの追加**』およびオンライン・ヘルプを参照してください。

プロセス・アクティビティについてのチェック

- プロセスはプロセス・アクティビティに割り当てる必要がありますが、ローカル・データベースにはなくても構いません。実行時バインディングの概念が適用されます。つまり、プロセスが存在しているかどうかのチェックは実行時にのみ行われます。
- 開始アクティビティが参照するプロセスが、その開始アクティビティが属しているプロセスと同じであってはなりません。入ってくる制御コネクタは、開始アクティビティには含まれません。他のアクティビティは、それぞれのプロセスを再帰的に呼び出すことができます。

ブロック・アクティビティーについてのチェック

- ダイアグラムを空にすることはできません。ダイアグラムの中には最低でも 1 つのアクティビティーを含める必要があります。
- プロセス内にあるブロック・アクティビティーの合計数が 32766 を超えてはなりません。
- ブロックの最大ネスト・レベルが 100 を超えてはなりません。

プログラム・アクティビティーについてのチェック

- プログラムは関係するプログラム・アクティビティーに割り当てる必要があります。また、そのプログラムは実際に存在していなければなりません。
- 関連プログラムの「**プログラム・プロパティー (Program properties)**」で「**プログラムはこれらのデータ構造を使用 (Program requires these data structures)**」を選択する場合、プログラム・アクティビティーの入力データ構造をプログラムの入力データ構造と同じにしなければなりません。同様に、プログラム・アクティビティーの出力データ構造も、関連プログラムの出力データ構造と同じにしなければなりません。
- 最低でも 1 つのプラットフォームのプログラム・プロパティーを定義する必要があります。使用できるプラットフォームは、Windows NT、Windows 9x、OS/2、AIX、HP-UX、または OS/390 です。
- 「**コンテナにあるプログラム実行サーバー (Program execution server From container)**」を定義する場合には、ストリング・タイプのデータ構造が存在していなければなりません。
- プログラム実行サーバー名 (PES) を指定した場合、「**プログラム・アクティビティー・プロパティー (Program activity properties)**」の「**実行ページ (Execution page)**」で同期実行モードが選択されているかどうかを検査されます。サーバーのタイプが完全に限定されていないために確認されない場合は、実行モードの選択は検査されません。
- プログラム実行サーバー (PES) のモードは、実行時に同期するよう設定されています。このことは、たとえば、入力コンテナがプログラム実行サーバー名 (PES または UPES) を決定してしまうため、モデリング中にモードを選択できない場合に必要となります。
- 関連プログラムで Windows NT、Windows 9x、OS/2、AIX、HP-UX、または Solaris 用の実行可能プログラムまたはライブラリー (DLL または共用ライブラリー) が使用されている場合、関連プラットフォームのコマンド行パラメーターで置換変数として使用されて

いるすべてのメンバーが、プログラム・アクティビティーの入力データ構造に存在していなければなりません。

- 関連プログラムで Windows NT、Windows 9x、OS/2、AIX、HP-UX、または Solaris 用の実行可能プログラムまたはライブラリー (DLL または共用ライブラリー) が使用されている場合、プラットフォームのエントリー・ポイントを設定する必要があります。
- 「**プログラム実行 (Program Execution)**」用に選択した定義が、次の条件を満たしていなければなりません。

「**プログラム実行サーバー (Program execution server)**」を定義する場合には、関連プログラムの「**プログラム・プロパティ (Program properties)**」で「**プログラムの不在実行可能 (Program can run unattended)**」を選択しなければなりません。

- 関連プログラムに OS/390 外部サービスが含まれている場合には、以下のプログラム・プロパティを設定する必要があります。
 - 「**サービス (Service)**」
 - 「**呼び出しタイプ (Invocation type)**」
 - 「**実行可能プログラム (Executable)**」
 - 「**実行可能プログラム・タイプ (Executable type)**」

これらのプロパティを作成または変更するには、「**プログラム・アクティビティー・プロパティ (Program activity properties)**」ウィンドウをオープンします。プログラム・プロパティを定義するには、ツリー・ビューで「**インプリメンテーション**」タブをクリックし、「**プログラム・プロパティ (Program properties)**」ウィンドウをオープンします。その後、変更したいデータのタイプに関するタブをクリックします。

制御コネクタについてのチェック

- 分岐条件は有効なブール式でなければなりません。有効なブール式については、81ページの『条件の構文』で説明されている条件構文を参照してください。使用するデータ構造のメンバーはすべて、出力データ構造の中に存在していなければなりません。また、それらのメンバーの中には、使用されるコンテキストに応じて適切なタイプが含まれていなければなりません。コネクタのソース・アクティビティー以外のアクティビティーからのデータ構造メンバーを使用する場合は、そのアクティビティーからコネクタのソース・アクティビティーまでの制御パスが存在していなければなりません。

制御コネクターを定義する方法についての詳細は、43ページの『プロセス・フローを制御するコネクター』およびオンライン・ヘルプを参照してください。

データ・コネクターについてのチェック

- 「**データ・マッピング (Data Mapping)**」ウィンドウでは、すべてのデータ・マッピングの**起点**メンバーが、ソース・データ構造、すなわちソース・アクティビティの出力データ構造に存在していなければなりません。ただし、コネクターが**ソース・ノード**から始まっている場合は別です。コネクターが**ソース・ノード**から始まっている場合は、親ブロック・アクティビティまたはプロセスの入力データ構造が適用されます。

注: 「**ターゲット・データ構造 (Target Data Structure)**」ペインの「**マッピング列 (Mapping column)**」には、実際に使用されている**起点**メンバーしか表示されません。

- すべてのデータ・マッピングの**宛先**メンバーが、**ターゲット・データ構造**、すなわち**ターゲット・アクティビティ**の入力データ構造に存在していなければなりません。ただし、コネクターが**シンク・ノード**で終わっている場合は別です。コネクターが**シンク・ノード**で終わっている場合は、親ブロック・アクティビティまたはプロセスの出力データ構造が適用されます。

注: 「**ターゲットデータ構造 (Target Data Structure)**」ペインの「**メンバー列 (Member column)**」には**宛先**メンバーが表示されます。

- 事前定義されている読み取り専用メンバー、すなわち `_PROCESS`、`_PROCESS_MODEL`、`_ACTIVITY`、`_RC` を**宛先**メンバーとして使用することはできません。このことはすべてのデータ・マッピングにあてはまります。
- データ・コネクターの**ソース・アクティビティ**は、制御パスを使用して**ターゲット・アクティビティ**に接続する必要があります。
- 各データ・マッピングの**起点**メンバーと**宛先**メンバーは、同じタイプでなければなりません。`_PROCESS_INFO` を `_PROCESS_INFO` にマッピングしたり、`_ACTIVITY_INFO` を `_ACTIVITY_INFO` にマッピングしたりすることもできます。この事前定義メンバーにはタイプがないことに注意してください。

注: 定義済みのデータ・マッピングがデータ・コネクターにないと、警告が出されます。

データ・コネクタを定義する方法についての詳細は、44ページの『データ・フローを制御するコネクタ』およびオンライン・ヘルプを参照してください。

データ構造についてのチェック

- データ構造にループを含めることはできません。データ構造 A にデータ構造 B がメンバー・タイプとして含まれている場合、正常に機能しません。

データ構造を定義する方法についての詳細は、35ページの『データ構造の定義』およびオンライン・ヘルプを参照してください。

定義機能からの搬出

定義機能搬出ユーティリティは、このデータベースから ASCII テキスト・ファイルへ定義を搬出する機能を備えています。搬出されたテキスト・ファイルは、ワークフロー定義言語 (FDL) という形式です。FDL の構文については、73ページの『第6章 FDL ファイル内でのワークフロー情報の定義』に記載されています。ワークフロー定義を定義機能から実行機能へ転送するには、62ページの『実行機能搬出および搬入ユーティリティの使用』に従って、まず定義機能搬出ユーティリティを使用し、それから実行機能搬入ユーティリティを使用して FDL を実行機能データベースに搬入します。



ユーザー定義アイコンは、定義機能から自動的に搬出されません。自分で作成したユーザー定義アイコンを定義機能と実行機能の両方で使用したい場合には、*IBM MQSeries Workflow: インストールの手引き* で説明されている方法でアイコンをインストールしなければなりません。

定義機能搬出の開始と使用

定義機能搬出ユーティリティを開始および使用するには、以下のようにします。

1. メニュー・バーで「**定義機能**」をクリックします。
2. 「**搬出**」をクリックします。
これにより、「**搬出**」ダイアログがオープンします。
3. 搬出する項目を選択し、「**了解**」をクリックして搬出を開始します。

デフォルトでは、定義機能データベース内のすべての定義が表示されます。オブジェクトを選択することによって、定義のリストをフィルターに掛けることができます。必要な搬出形式を選択することができます。ワーク・フロー定義を実行機能に搬入するには、FDL の形式を選択する必要があります。FDL 形

式の代わりに HTML を選択することもできます。たとえば、文書として印刷したい場合などには、HTML を選択することができます。

詳細については、オンライン・ヘルプを参照してください。

定義機能へ搬入

FDL ファイルから定義を定義機能データベースへ搬入するには、定義機能搬入ユーティリティを使用することができます。定義機能にワークフロー情報を搬入して、以下のことを行うことができます。

- 定義機能データベースの内容を復元する。
- 実行機能データベースの内容と同期化する。
- MQSeries Workflow の外で作成された定義を搬入する。

データベースの同期化についての詳細は、8ページの『定義機能と実行機能が協働する方法』を参照してください。



定義機能に Microsoft Jet データベース・エンジンを使用している場合には、FDL ファイルを搬入する際の名前は固有でなければなりません。

たとえば、役割、プロセス、データ・ストリング、およびプログラム登録の名前を定義する場合は、大文字と小文字の区別もしなければなりません。たとえば、*program1* を定義して、それから別の名前 *PROGRAM1* を定義することはできません。

定義機能搬入の開始と使用

定義機能搬入ユーティリティを開始および使用するには、以下のようになります。

1. メニュー・バーで「**定義機能**」をクリックします。
2. 「**搬入**」をクリックします。
これにより搬入ダイアログがオープンします。
3. 搬入する項目を選択し、「**了解**」をクリックして搬入を開始します。

FDL ファイルの出所が実行機能の場合には、「**実行機能からの FDL (FDL from Runtime)**」をクリックします。



定義機能への搬入で既存のオブジェクトを上書きするのを避けるために、「上書き」を選択しないでください。

DELETE キーワードが含まれる FDL ファイルを定義機能に搬入する場合は、以下の事柄が適用されます。

- FDL ファイルが実行機能から搬出されたファイルである場合、オブジェクトはデータベース内で削除されます。
- FDL が他のシステムに由来している場合、オブジェクトは削除のためにマークされます。

詳細については、オンライン・ヘルプを参照してください。

実行機能搬出および搬入ユーティリティーの使用

実行機能搬出および搬入ユーティリティーを使用すると、以下のことが実行できます。

- 実行機能データベースから FDL ファイルにワークフロー定義を搬出する。
- FDL ファイルを実行機能データベースに搬入する。



ユーザー定義アイコンは、搬出および搬入ユーティリティーの一部ではありません。自分で作成したユーザー定義アイコンを定義機能と実行機能の両方で使用したい場合には、*IBM MQSeries Workflow: インストールの手引き* で説明されている方法でアイコンをインストールしなければなりません。

搬出および搬入ユーティリティーは独立型ユーティリティーであり、MQSeries Workflow サーバーでコマンド・プロンプトから開始されます。

ユーティリティーを使用して、以下のことを実行できます。

- 定義機能からのワークフロー定義を使用して、実行機能に新しいデータベースを作成する。
- 定義機能からワークフロー定義を搬入して変換する。
- MQSeries Workflow の外で作成した FDL ファイルを搬入する。
- 実行機能データベースから FDL ファイルを搬出する。
- FDL ファイルを搬入および検査する。

実行機能搬出 / 搬入ユーティリティーの開始

ユーティリティーは、以下の 2 通りのモードで開始できます。

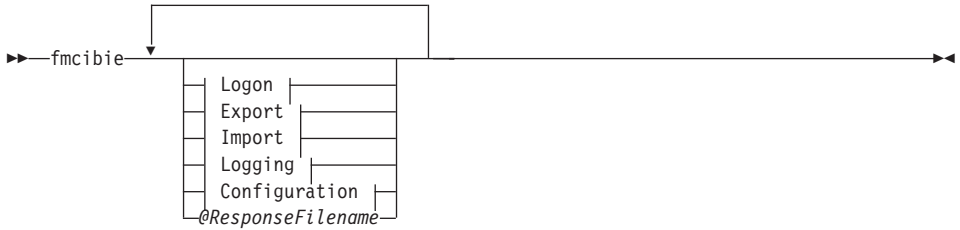
- 搬入モード
- 搬出モード

ユーティリティーを開始するには、MQSeries Workflow サーバーにログオンし、以下のことを実行します。

1. Windows NT または AIX では、コマンド・プロンプトをオープンし、MQSeries Workflow がインストールされているディレクトリーに移動します。
2. コマンド・プロンプト・ウィンドウで、以下のいずれかを入力します。
 - **fmcibie /i=in.fdl**
これによりユーティリティーが開始され、ファイル名 *in.fdl* の FDL ファイルを搬入します。
 - **fmcibie /e=out.fdl**
これによりユーティリティーが開始され、ファイル名 *out.fdl* の FDL ファイルに定義を搬出します。

以下の構文図は、ユーティリティーを使用する方法を示しています。

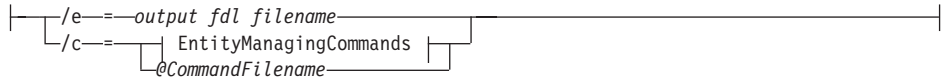
搬出および搬入ユーティリティのコマンド構文



Logon:



Export:



Import:



Logging:



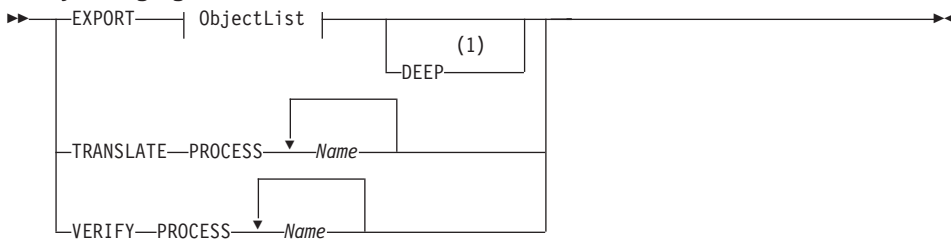
Configuration:



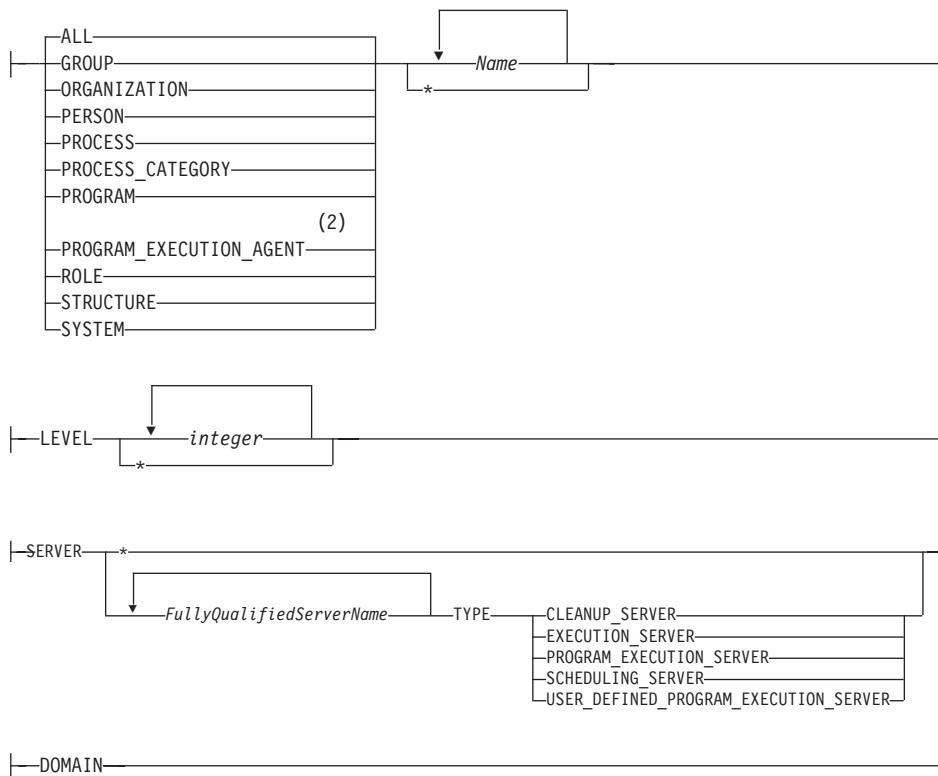
注:

- 1 構成識別子には、最大 8 文字の英数字を指定できますが、DBCS 文字は使用できません。

EntityManagingCommands



ObjectList:



注:

- 1 処理に関しては、DEEP のみ有効です。
- 2 PROGRAM_EXECUTION_AGENT の名前は、RELATED_PERSON 属性の *PersonName* です。

注:

- オプションと可能な引き数の区切り文字としては、等号 (=)、コロン (:)、またはブランク文字を使用できます。
- これらのオプションはどれも 1 度しか指定できません。
- オプション **i** またはオプション **c** のいずれかを使用できますが、両方を指定することはできません。
- 搬出ログ・ファイルのファイル名の拡張子は、**LOG** です。
- ログのデフォルト宛先は、stderr (cerr) です。
- オプション **c** には、引用符で囲んで複数の語を使用することができます。

搬出 / 搬入ユーティリティーのオプション

以下のオプションを使用して、搬出 / 搬入ユーティリティー (fmcibie) を開始できます。

オプション	引き数	説明
c	Command string	これは、搬出モードでのみ、またはファイル名が単価記号 (@) で始まる場合のみ、エンティティー・コマンドを管理します。指定するファイルには、使用したいコマンドが含まれています。
e	output FDL filename	データベースから、指定したエンティティーを、指定した出力 FDL ファイル名に搬出します。搬出モードの場合のみ使用されます。
h		ユーティリティーのヘルプ情報を表示します。
i	import FDL filename	指定したエンティティーを、指定した搬入 FDL ファイルからデータベースに搬入します。
l	log filename	通知、警告、およびエラー・メッセージが書き込まれる宛先を指定します。このオプションを指定しない場合の省略宛先は stderr です。このオプションにファイル名を指定しないなら、入出力 FDL ファイル名にファイル拡張子として LOG を付けたログ・ファイル名が作成されます。
o		既存のデータベース・エンティティーを上書きします。ただし、搬入モードのみです。

オプション	引き数	説明
f		以下の規則に従い、インポート・アクションを強制します。 <ul style="list-style-type: none"> 参照されているオブジェクトが存在している場合、CREATE アクションは REPLACE アクションとして実行されます。これは上書き (o) オプションに相当します。 参照されているオブジェクトが存在しない場合は、UPDATE/REPLACE アクションは CREATE アクションとして実行されます。DELETE アクションは無視されません。
p	password	指定したユーザー ID のパスワードです。インストールに統一ログオンを使用する場合には、パスワードは任意選択です。統一ログオンは、Windows NT および UNIX でのみ可能です。しかし、ユーザー ID を指定する場合は、即座にパスワードを入力する必要があります。ご注意ください。
t		プロセス・モデルを変換します。ただし、搬入モードのみです。プロセス・モデルの変換には、プロセス・モデルの検査も含まれません。
u	userid	これは MQSeries Workflow のログオン・ユーザー ID です。 p オプションで説明されているとおり、インストール・システムに統一ログオンを使用する場合には、ユーザー ID は任意選択です。
v		これにより、51ページの『ワークフロー・モデルの検証』に従って、プロセス・モデルが検査されます。
y	instance	MQSeries Workflow インスタンス名が、プロファイル設定にアクセスするために使用されます。

これらのオプションを個々に指定する代わりに、応答ファイルを使用して、いくつかのオプションを含めることができます。応答ファイルを使用するには、次のように指定します。

`fmcibie @ResponseFilename`

応答ファイルは次のように使用します。

1. テキスト・ファイルを作成します。例、`myfile.text`
2. 1 つまたは複数のオプションを使用します。そのさい、各コマンド行パラメーターを 1 行に 1 つずつ指定します。たとえば、次のようにします。

```
/i=test.fdl  
/u=ADMIN  
/p=password  
/o  
/t
```

これは、コマンド行パラメーターとしては指定せずに、ファイルのパスワードを指定しようとする場合に特に便利です。

オプションの搬入方法の例については、『搬入の例』を参照してください。UNIX 環境では、スラッシュ (/) の代わりにマイナス (-) を使用してオプションを指定する必要があることにご注意ください。

搬出 / 搬入ユーティリティーのエラー・コード

ファイルを搬出または搬入しているときに搬出 / 搬入ユーティリティーがエラーを検出すると、戻りコードが指定されます。戻りコードの値が 2 より大きい場合には、ユーティリティーは停止します。重大エラーの結果として、いわゆるトランザクションのロールバックが実行されます。つまり、データベースは変更されません。

表 4. 搬出 / 搬入ユーティリティーのエラー・コード

値	説明
0	OK - エラーはありません
1	通知メッセージです
2	警告メッセージです
4	検証エラー
8	構文エラー - ユーティリティーは停止します
12	エラー・メッセージです
16	入力エラー
20	重大エラー
24	内部エラー

搬入の例

搬入オプションの使用例を以下に示します。

FDL ファイルを搬入する

```
fmcibie /i=in.fdl /u=admin /p=pwd
```

これによりユーティリティーが開始され、ファイル名 *in.fdl* の FDL ファイルを搬入します。ユーザー ID *admin* およびパスワード *pwd* でログオンします。

プロセス・モデルを搬入して変換する

```
fmcibie li=in.fdl lu=admin lp=pwd lt
```

これによりユーティリティーが開始され、実行機能で使用するために FDL ファイルを搬入および変換します。

搬入してログ・ファイルにメッセージを書き込む

```
fmcibie li=in.fdl lu=admin lp=pwd llog1.log
```

これによりユーティリティーが開始され、FDL ファイルを搬入し、ファイル名 *log1.log* のログ・ファイルに情報が書き込まれます。

搬出の例

搬出オプションの使用例を以下に示します。

FDL ファイルを搬出する

```
fmcibie le=out.fdl lu=admin lp=pwd
```

これによりユーティリティーが開始され、ファイル名 *out.fdl* の FDL ファイルを搬出し、ユーザー ID *admin* およびパスワード *pwd* でログオンします。

すべての人を搬出する

```
fmcibie le=out.fdl lu=admin lp=pwd lc"EXPORT PERSON*"
```

これにより、ユーティリティーが開始され、ワークフロー・モデルのすべての人の定義を搬出します。

特定の人を搬出する

```
fmcibie le=out.fdl lu=admin lp=pwd lc"EXPORT PERSON 'ERIC'  
'TOM'"
```

これにより、ユーティリティーが開始され、ERIC および TOM の定義を搬出します。

コマンド・ファイルからコマンドを搬出して使用する

```
fmcibie le=out.fdl lu=admin lp=pwd lc@test1
```

これにより、ユーティリティーが開始し、FDL ファイルを搬出し、ファイルからコマンド (たとえば、*test1*) を使用します。たとえば、ファイル *test1* は次のようにすることができます。

```
EXPORT DOMAIN  
EXPORT SERVER *
```

変換の例

既存のモデルを変換する

```
fmcibie /u=admin /p=pwd /c"TRANSLATE PROCESS process1"
```

これによりユーティリティーが開始され、プロセス名 *process1* で実行機能データベースの既存のプロセス・モデルを変換します。

MQSeries Workflow バージョン 3.1x のワークフロー・モデルをバージョン 3.2.2 で使う

このリリースで、MQSeries Workflow の旧リリースのいずれかで作成されたワークフロー・モデルを使用したい場合には、以下のことを実行しなければなりません。

1. MQSeries Workflow の新しいリリースをインストールする前に、60ページの『定義機能からの搬出』に従って、定義機能データを搬出します。「搬出」ウィンドウの、「すべて搬出 (Export all)」および「FDL」を選択したことを確認してください。
2. *IBM MQSeries Workflow: インストールの手引き* に従って、MQSeries Workflow の新しいリリースをインストールします。
3. ステップ 1 の FDL ファイルを搬入します。FDL の搬入方法については、61ページの『定義機能へ搬入』を参照してください。

これで、MQSeries Workflow の新しいリリースのワークフロー・モデルを使用することができます。

FlowMark バージョン 2.3 FDL のワークフロー・モデルの使用

FlowMark バージョン 2.3 で作成した FDL ファイルを使いたい場合は、次のようにする必要があります。

1. FlowMark バージョン 2.3 の FDL を定義機能に搬入します (61ページの『定義機能へ搬入』を参照)。
2. この更新されたバージョンの FDL ファイルを搬出します (60ページの『定義機能からの搬出』を参照)。
3. FDL ファイルを実行機能に搬入します (62ページの『実行機能搬出および搬入ユーティリティーの使用』を参照)。

第2部 MQSeries Workflow の外部形式の使用

第6章 FDL ファイル内でのワークフロー情報

の定義	73
構文図の読み方	73
FDL の構文規則	75
サイズ制限	76
名前と文字列の構文規則	76
ActivityName	76
Codepage	76
Description と Documentation	76
EnvironmentString	76
ExternalContextString	77
ExternalShortString	77
ExternalString	77
FileName	77
Float	77
FullyQualifiedActivityName	77
Level	78
Long	78
MappingString	78
MemberName	78
ModLevel	78
MQSeriesObjectName	78
MQSeriesQueueManagerName	78
Name	79
ObjectName	79
ObjectShortName	79
ParameterString	79
PasswordString	79
PersonName	79
Priority	80
ProcessName	80
String	80
SymbolName	80
SystemQualifier	80
WorkingDirectory	81
条件の構文	81
ブール式 (Boolean expression)	81
比較演算子 (Comparison operator)	81
整数式 (Integer expression)	82
数式 (Numeric expression)	82
2 進数式 (Binary expression)	82

文字列式 (String expression)	83
ContainerMember	83
スコープ (Scope)	83
DataStructureMemberName	83
DottedName	84
ProcessInfoMember	84
ActivityInfoMember	84
終了条件および分岐条件の表記	84
条件の評価	87

第7章 FDL 定義 89

FDL ソース・ファイルの形式	89
FDL ソース・ファイル	89
トポロジ (Topology)	92
ドメイン (Domain)	92
システム・グループ (System Group)	92
システム (System)	93
TopologySetting	93
OperationSetting	94
SessionSetting	95
DefaultServerSetting	95
ExecutionServerContext	95
CleanupServerContext	95
ProgramExecutionServerContext	96
UPESContext	96
SchedulingServerContext	96
DefaultProgramExecutionAgentSetting	96
DefaultProcessSetting	97
自律 (Autonomy)	97
DefaultActivitySetting	97
DefaultProgramSetting	98
DefaultImportSetting	98
サーバー (Server)	98
ProgramExecutionAgent	99
QueueManager	99
スタッフ (Staff)	100
人 (Person)	100
役割 (Role)	101
組織 (Organization)	101
レベル (Level)	102
プロセス・モデル定義 (Process Modeling)	102

データ構造 (Data structure)	102
プログラム (Program)	103
ProgramSetting	104
PlatformSetting	104
UNIXSetting	104
WindowsSetting	105
OS/2Setting	105
DLLSetting	105
EXESetting	105
EXTERNALSetting	106
プロセス (Process)	106
ProcessSetting	107
ProcessStaffAssignmentSetting	107
ProcessGraphicsSetting	108
構成 (Construct)	108
アクティビティ (Activity)	108
ProgramActivity	108
ProcessActivity	109
ブロック (Block)	109
ActivitySetting	110
ActivityExtensionSetting	111
ActivityStaffAssignmentSetting	111
通知 (Notification)	112
ControlFlow	113
DataFlow	113
プロセス・カテゴリー (Process category)	114
ツール・セット (ToolSet)	114
共通変数	115
ScreenPosition	115
SymbolLayout	115
ContainerLayout	115
WindowLayout	116
ContainerInitial	116
BendPoints	116
カラー (Color)	116
ColorSetting	117
TextSettings	117
FontSettings	117
TimeStamp	117
TimeInterval	118
TimePeriod	118
TimeEvent	118
MessageLength	118
FullyQualifiedServerName	119

第6章 FDL ファイル内でのワークフロー情報の定義

ファイルの中にワークフロー情報を定義したなら、それを MQSeries Workflow 定義機能に搬入することになります (49ページの『定義機能および実行機能でのワークフロー定義の使用』を参照)。

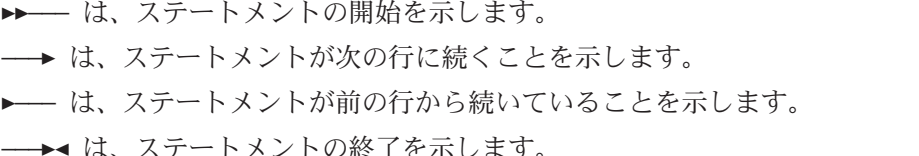


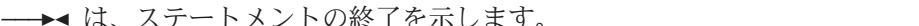




この章では、FDL ソース・ファイル内の宣言およびプロセス定義の構文について説明します。

構文図の読み方

本書では、FDL のプログラミング構文を説明するために、ダイアグラムを使用しています。ダイアグラムは、左から右に、上から下に読んでください。これらのダイアグラムでは、すべてのスペースおよび文字に意味があります。

それぞれのダイアグラムは右矢印の先が 2 つ重なった記号で始まり、右矢印と左矢印の先頭の組で終わります。

本書で使用されている構文図には、以下の規則が適用されます。

-  は、ステートメントの開始を示します。
 -  は、ステートメントが次の行に続くことを示します。
 -  は、ステートメントが前の行から続いていることを示します。
 -  は、ステートメントの終了を示します。完全なステートメント以外の構文単位のダイアグラムは、 記号で始まり、 記号で終わります。
- 必須項目は、メインパスの水平線上に現れます。
- オプション項目は、通常、メイン・パスの下に現れます。

メイン・パスの上にオプション項目が現れる場合、その項目はステートメントの実行には影響を及ぼさず、可読性の目的でのみ使用されます。



- 2 つ以上の項目から選択することができる場合、縦方向のスタック中に現れます。

項目の 1 つを選択する必要がある場合、スタック中の 1 項目がメインパス上に現れます。



1 つを選択する項目が任意選択項目である場合、スタック全体がメインパスの下に現れます。



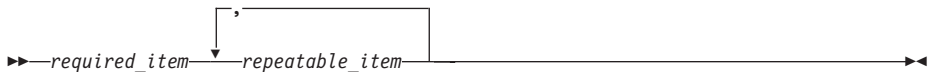
項目のいずれかがデフォルトの場合には、これがメイン・パスの上に現れ、残りの項目は下に現れます。



- 主線上の、左へ戻る矢印は、繰り返すことができる項目を示します。



反復矢印にコンマが含まれる場合、反復される項目をコンマで区切る必要があります。



繰り返し矢印に括弧で囲まれた数字が含まれている場合には、その数字は、項目を繰り返せる最大の回数を示しています。



スタックの上の繰り返し矢印は、スタックの項目を繰り返せることを示します。

- キーワードは大文字で表示されます (たとえば、FROM)。変数はイタリックで表示されます (たとえば、*column name*)。これらは、ユーザーが指定する名前または値です。
- 句読記号、括弧、算術演算子、その他の記号が表示されている場合は、それらを構文の一部として入力しなければなりません。
- 構文図は、フラグメントに分割することができます。フラグメントは縦線によって区切られ、縦線の上にフラグメントの名前が入ります。フラグメントは、下記のようにして主構文図の後に示されます。

```
| A Fragment |-----|
```

フラグメント (A Fragment):

```
| Keyword—value-----|
```

FDL の構文規則

続くセクションでは、定義機能ウィンドウのフィールドに値を入力したり、独自の FDL ファイルを作成したりするときに従う必要のある規則について説明します。プロセス・モデルを検証することについての詳細は、51ページの『ワークフロー・モデルの検証に関する規則』を参照してください。

名前と文字列に関する規則を以下に示します。

属性を複数回指定する場合

同じ属性を複数回指定した場合は、最後の定義が有効になります。例外がある場合は、その旨を明示的に示します。

引用符 二重引用符で囲まれた文字列の中には、単一引用符を組み込むことができます。単一引用符で囲まれた文字列の中には、二重引用符を組み込むことができます。そのようにして 1 個の引用符を含めるには、2 個連続して入力する必要があります。

名前 英字 (大文字または小文字)、数字、および下線文字以外の文字を含む名前は、引用符で囲む必要があります。名前が FDL キーワードと同じ場合も、その名前は引用符で囲む必要があります。

サイズ制限

FDL オブジェクトには、下記のようなサイズの制限があります。

- プログラム宣言: 30720 バイト
- 構造宣言: 30720 バイト
- プロセス宣言: 4190 KB

名前と文字列の構文規則

プロセス・モデルを実行時に正常に実行させるためには、MQSeries Workflow オブジェクトの命名規則を把握しておく必要があります。FDL 定義についての詳細は、89ページの『FDL ソース・ファイルの形式』を参照してください。名前と文字列に関する構文規則を以下に示します。

ActivityName

SymbolName と同じ規則が適用されます。

Codepage

指定するコード・ページ番号は、インストール済みシステムで有効かつ使用可能なものでなければなりません。

Description と Documentation

- *Description* には最大 254 文字まで指定することができます。*Documentation* には最大 4096 文字まで指定することができます。
- 制御文字を除く、あらゆる種類の文字を指定することができます。行終了文字 (CR、LF) を指定することもできます。
- 二重引用符で囲まれた文字列の中には、単一引用符を組み込むことができます。単一引用符で囲まれた文字列の中には、二重引用符を組み込むことができます。ただし、中に組み込んだ引用符を繰り返し使用することはできません。

EnvironmentString

- 最大 1024 文字 (バイト) を指定できます。
- 制御文字と & < > ~ / を除くあらゆる種類の文字を指定できます。
- 環境変数は [variable [string]] という形式です。
詳細については、オペレーティング・システムの資料を参照してください。

ExternalContextString

最大 32 文字まで指定することができます。所定の規則を 1 つでも満たしていない場合、このストリングは無効になります。このストリングに適用される規則を以下に示します。

- 最低 1 文字 (バイト)、最大 32 文字 (バイト) を指定できます。
- 大文字だけしか使用できません。使用できる文字は、英字 (0~9、A~Z) と \$ # @ の文字です。
- 最初の 3 文字を SYS にすることはできません。

ExternalShortString

最大 8 文字まで指定することができます。所定の規則を 1 つでも満たしていない場合、このストリングは無効になります。このストリングに適用される規則を以下に示します。

- 最低 1 文字 (バイト)、最大 8 文字 (バイト) を指定できます。
- 使用できるのは大文字だけです。使用できる文字は、英数字 (0~9、A~Z) と \$ # @ , - の文字です。
- 最初の文字は A~Z \$ # @ のいずれかにしなければなりません。

ExternalString

- 最低 1 文字 (バイト)、最大 32 文字 (バイト) を指定できます。
- 制御文字と DBCS (2 バイト文字セット) 文字を除くあらゆる種類の文字を指定できます。

FileName

- 最大 254 文字 (バイト) を指定できます。
- 名前は、有効なファイル名または完全修飾ファイル名でなければなりません。
詳細については、オペレーティング・システムの資料を参照してください。

Float

- 最低 1 桁 (バイト)、最大 15 文字 (バイト) を指定できます。
- 浮動小数点数の場合、整数部分と小数部分の区切りの小数点文字としては、ピリオド (.) を使う必要があります。

FullyQualifiedActivityName

- 最大 254 文字 (バイト) を指定できます。
- ピリオド (.) で区切られた、有効な *SymbolNames* から成り立っていなければなりません。

Level

- 数字 1 文字を指定できます。有効な値は 0~9 です。

Long

- 最低 1 桁 (バイト)、最大 10 文字 (バイト) を指定できます。

MappingString

- 最低 1 文字 (バイト)、最大 254 文字 (バイト) を指定できます。
- 制御文字と DBCS 文字を除くあらゆる種類の文字を指定できます。

MemberName

- *MemberName* の先頭を下線文字にすることはできません。
- *SymbolName* と同じ規則が適用されます。

ModLevel

- MQSeries Workflow リリースの修正レベルを指定する整数。

MQSeriesObjectName

構文は次のものに適用されます。

- OS/2 WARP 4.0
- Windows NT 4.0
- AIX 4.2
- 最大 48 文字まで指定することができます。
- 使用できるのは、英数字 0~9 と A~Z (大文字または小文字)、および / % の文字です。ただし / と % は特殊文字なので、二重引用符で囲まなければなりません。
- 先行ブランクまたは組み込みブランクは使用できません。
- 各国語文字は使用できません。
- 名前は二重引用符で囲むことができますが、それは名前に特殊文字が含まれている時のみ意味を持ちます。

MQSeriesQueueManagerName

構文は次のものに適用されます。

- OS/2 WARP 4.0
- Windows NT 4.0
- AIX 4.2
- 最大 8 文字まで指定することができます。

- 使用できるのは、英数字 0～9 と A～Z (大文字または小文字)、および / % の文字です。ただし / と % は特殊文字なので、二重引用符で囲まなければなりません。
- 先行ブランクまたは組み込みブランクは使用できません。
- 各国語文字は使用できません。
- 名前は二重引用符で囲むことができますが、それは名前に特殊文字が含まれている時のみ意味を持ちます。

Name

- 最大 32 文字まで指定することができます。
- 制御文字を除く、あらゆる種類の文字を指定することができます。

ObjectName

- 最低 1 文字 (バイト)、最大 32 文字 (バイト) を指定できます。
- 制御文字を除く、あらゆる種類の文字を指定することができます。

ObjectShortName

- 最低 1 文字 (バイト)、最大 8 文字 (バイト) を指定できます。
- 使用できるのは英数字 0～9、A～Z (大文字または小文字) です。
- 制御文字を除く、あらゆる種類の文字を指定することができます。

ParameterString

- 最大 256 文字 (バイト) を指定できます。
- 制御文字を除く、あらゆる種類の文字を指定することができます。

PasswordString

最大 32 文字まで指定することができます。以下の規則が適用されます。

- 長さは 32 文字 (バイト) 以下です。
- 制御文字を含めることはできません。
- DBCS 文字を含めることはできません。
- 日本語 SBCS 文字 (半角カタカナ) を含めることはできません。

PersonName

- 最大 32 文字まで指定することができます。
- @ < > [] ¥ “ ; などの文字を含めることはできません。
- 制御文字を含めることはできません。
- DBCS 文字を含めることはできません。

- ローカル環境の小文字を含めることはできません。
- 現在環境の英字 (大文字)、数字、句読文字を指定するか、またはブランク ‘ ’ とします。

Priority

- 数字 1 文字を指定できます。有効な値は 0~9 です。

ProcessName

- 最大 63 文字 (バイト) を指定できます。
- 現在のロケールに応じて、どの印刷可能文字でも使用できます。ただし、次のものは使用できません。 * ? " ; : . \$
- ブランクの使用には制限があります。つまり、先行ブランク、末尾ブランク、および連続ブランクは使用できません。

String

- あらゆる種類の文字を指定することができます。
- 二重引用符で囲まれた文字列の中には、単一引用符を組み込むことができます。単一引用符で囲まれた文字列の中には、二重引用符を組み込むことができます。ただし、中に組み込んだ引用符を繰り返し使用することはできません。

条件の中にストリングを使用することについては、84ページの『終了条件および分岐条件の表記』の中の『ストリング』の部分を参照してください。

SymbolName

- 最大 32 文字まで指定することができます。
- 次の文字を含めることはできません。 ! ‘ [] * + , - . ; / : < = > () ¥ ^ “
- 次のキーワードは使えません。
AND、IS、LOWER、MOD、NOT、NULL、OR、SUBSTR、UPPER、VALUE、または特殊名 _BLOCK および _STRUCT
- 先頭に数字は使えません。
- 制御文字を含めることはできません。
- 先行ブランク、末尾ブランク、または連続ブランクを含めることはできません。

SystemQualifier

- 最大 8 文字まで指定することができます。
- 使用できるのは英数字 0~9、A~Z (大文字または小文字) です。

WorkingDirectory

- 最大 254 文字 (バイト) を指定できます。
- 名前は、有効なディレクトリー名でなければなりません。
詳細については、オペレーティング・システムの資料を参照してください。

条件の構文

以下の例は、条件を表す論理式のコーディング方法を示したものです。

条件:

▶ | Boolean expression | ◀

ブール式 (Boolean expression)

Boolean expression	AND	Boolean expression
Boolean expression	OR	Boolean expression
NOT		Boolean expression
	(1)	
String expression	Comparison operator	String expression
Numeric expression	Comparison operator	Numeric expression
	(2)	
Binary expression	Comparison operator	Binary expression
Integer expression		
	(3)	
ContainerMember	IS	NULL
	(4)	
ContainerMember	NOT	NULL
(Boolean expression)

注:

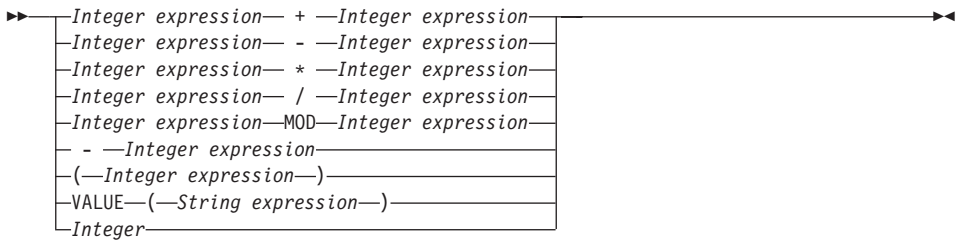
- 1 入力された ASCII 文字コードの値に基づいて、ストリングが 1 文字 1 文字比較されます。
- 2 有効な比較演算子 (Comparison operator) は = と <> だけです。
- 3 この演算子は、コンテナ・メンバーが設定されていないことを調べるために使います。
- 4 この演算子は、コンテナ・メンバーが設定されていることを調べるために使います。

比較演算子 (Comparison operator)

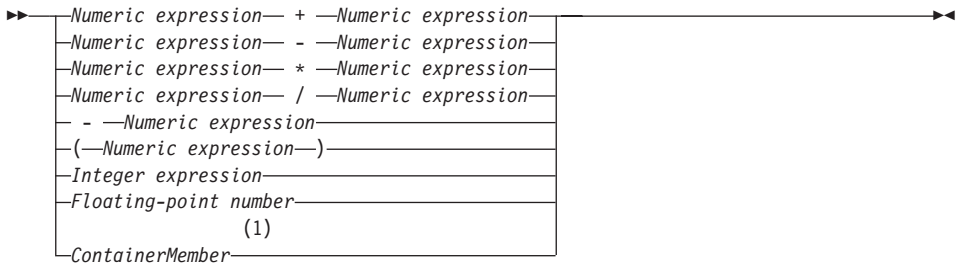


比較の両辺が NULL なら結果は不明 (unknown) になります。

整数式 (Integer expression)



数式 (Numeric expression)



注:

1 メンバーは Long 型または Float 型でなければなりません。

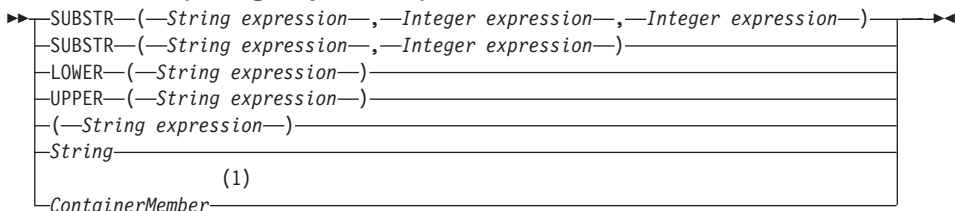
2 進数式 (Binary expression)



注:

- 1 2進数リテラルでは、各1バイトが2桁の16進数で表記されます。
たとえば、復帰改行 (CRLF) は "0D0A" で表されます。
- 2 メンバーは Binary 型でなければなりません。

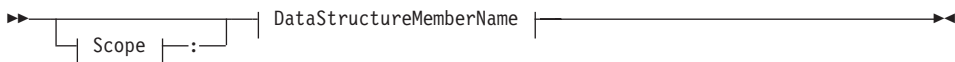
ストリング式 (String expression)



注:

- 1 メンバーは String 型でなければなりません。

ContainerMember



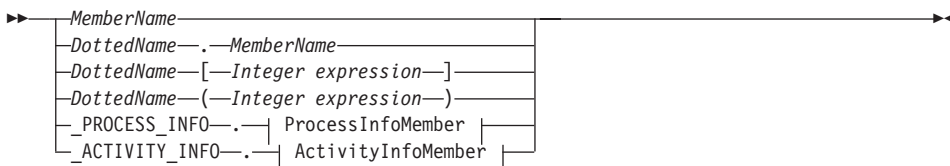
スコープ (Scope)



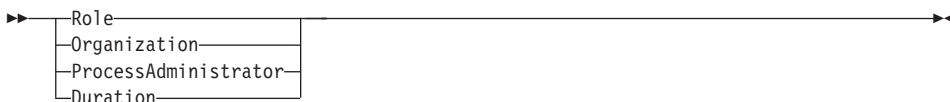
DataStructureMemberName



DottedName



ProcessInfoMember



ActivityInfoMember



終了条件および分岐条件の表記

終了条件と分岐条件を定義する際には、以下の表記を使用します。

• 条件の中で使用される関数

これらの機能を指定するためのキーワードには、大文字小文字の区別がありません。条件で使用する機能を以下に示します。

LOWER

文字列を大文字から小文字へと変換する機能。たとえば、以下の例は真として評価されます。

```
LOWER("A") = "a"
```

UPPER

文字列を小文字から大文字へと変換する機能。たとえば、以下の例は真として評価されます。

```
UPPER("a") = "A"
```

SUBSTR

サブstring抽出をサポートする関数。たとえば、次のようなものです。

```
SUBSTR("abcde", 2,3)="bcd"
```

VALUE

stringの数値表記を対応する数値に変換する機能。たとえば、以下の例は真として評価されます。

```
VALUE("324")=324
```

当然のことながら、前述の例で用いられているリテラル値の個所に、データ・コンテナーからの変数を使用することもできます。

• 特殊名

_RC アクティビティの出力コンテナーにおけるリターン・コード。たとえば、終了条件や分岐条件として以下の式を使用することができます。

```
_RC=0
```

これは長整数です。

_STRUCT

デフォルトのデータ構造全体またはユーザー定義のデータ構造を参照します。たとえば、アクティビティ A の出力コンテナーの中身は次のように表すことができます。

```
A:_STRUCT
```

これは長整数です。

_BLOCK

現在のプロセスかプロセス内のブロックを参照します。たとえば、メンバー項目

```
ClientFound
```

の値を、現在ブロックのソース・コンテナーで次のようにテストすることができます。

```
_BLOCK:ClientFound="No"
```

当然のことながら、前述の例で用いられているリテラル値の個所に、データ・コンテナーからの変数を使用することもできます。

- **演算子**

以下のリストは、優先順位の高い方から低い方へと並べられています。同じ行に並べられている演算子は、優先順位が同じです。

NOT 単項ブール否定 (“NOT”) 演算子

- 単項算術負。

/ * 2 進算術演算子。

- + 2 進算術演算子。

> < = <= >= <>
2 進ブール演算子。

AND 2 進ブール演算子。

OR 2 進ブール演算子。

単項負と “NOT” 演算子を除き、どの演算子も左に関連付けられます。式の一部を囲んで操作の順序を指定するためには、(AND) のように括弧を使用してください。

- **NULL 演算子**

IS NULL と NOT NULL があります。特定のデータ構造メンバーが設定されているどうかを照会するのに、これらの演算子を使用してください。

- **メンバー名**

メンバー名に引用符は使用しません。次の文字を使用できます。 _ a~z
A~Z 0~9

上記以外の文字を使用する場合は、名前の全体を単一 引用符で囲む必要があります。これは、DBCS 文字が正しく解釈されるために必要です。単一引用符を使用した例を下記に示します。

'UpdateClient:Name.LastName'

- 条件式の中で名前を修飾するには、条件式が参照するアクティビティの名前の後にコロロン (:), その後に出力データ構成内のメンバー名を指定します。たとえば、

UpdateClient:Name

これは任意選択です。

- ネストされたデータ構造メンバーを表すためには、アクティビティ名の後に、コロロン、ネスト・データ構造の名前、ピリオド、データ構造メンバー項目の名前をこの順に指定します。たとえば、

UpdateClient:Name.LastName

- 配列に索引を付ける場合には、大括弧 [] を使用します。たとえば、次のようにします。

Addr.POBOX[0]

しかし、OS/390 サーバーで実行するプロセスでは、丸括弧 () を使用してください。

- 分岐条件の場合は、式の中の修飾されていない名前が、コネクタのソース・アクティビティの出力データ構造内にあるメンバー名を参照します。76ページの『名前と文字列の構文規則』を参照してください。
- 終了条件の場合は、修飾されていない名前が、終了条件が定義されているアクティビティの出力データ構造内にあるメンバー名を参照します。76ページの『名前と文字列の構文規則』を参照してください。

• 事前定義データ・メンバー

MQSeries Workflow で使用可能な事前定義データ・メンバーを使用することができます。これらの事前定義メンバーを利用するためには、コンテナ API を使用します。これらの事前定義メンバーの使用方法については、*IBM MQSeries Workflow: プログラミングの手引き* を参照してください。使用できる事前定義データ・メンバーのタイプは、以下のとおりです。

- 固定データ・メンバー
- プロセス情報データ・メンバー
- アクティビティ情報データ・メンバー

• 数値

これらは、10 進表記、8 進表記、または 16 進表記による、浮動小数点数または長整数 (32 ビット) です。整数の場合、先行 0 は 8 進数を示し、先行 0x または 0X は 16 進数を示します。

31	10 進数
037	8 進数
0x1f	16 進数
0X1F	16 進数

浮動小数を浮動小数点として使用することができます。

• スtring

String は、文字セット中の任意の文字の列を二重引用符で囲んだものです。String・リテラルに引用符文字が含まれている場合、この文字は他の引用符文字よりも先に来なければなりません。

条件の評価

条件が評価される際には、以下の規則が適用されます。

- **短絡評価。**条件評価部分のうち 1 つの部分だけでも条件全体の結果を判別できれば、条件を評価することはすぐに停止します。たとえば、次のようにします。

`(FirstName IS NULL) or FirstName="Melissa"`

データ構造メンバー `FirstName` が設定されていなければ、式 `FirstName IS NULL` が真として評価されます。結果として、条件全体が真として評価されます。2 番目の式は評価されません。

- **3 つの値による論理。**真 (True) と偽 (False) 以外に、不明 (*unknown*) と評価される条件もあります。たとえば、次のようにします。

`FirstName="Melissa" or (FirstName IS NULL)`

データ構造メンバー `FirstName` が設定されていないなら、式 `FirstName="Melissa"` は不明 (*unknown*) として評価されます。2 番目の式は真として評価されます。結果として、条件全体が真として評価されます。

ブール演算子 **AND**、**OR**、および **NOT** を使用した条件の真理値表については、表5 を参照してください。

表 5. **AND**、**OR**、および **NOT** 演算子の真理値表

		a AND b	a OR b	NOT a
a=t	b=t	t	t	f
	b=f	f	t	-
	b=?	?	t	-
a=f	b=t	f	t	t
	b=f	f	f	-
	b=?	f	?	-
a=?	b=t	?	t	?
	b=f	f	?	-
	b=?	?	?	-

凡例:

f	偽
t	真
?	不明

第7章 FDL 定義

この部分では、FDL ファイルを使ってワークフロー・モデルを定義する場合に使用できる FDL の構文について説明します。

FDL ソース・ファイルの形式

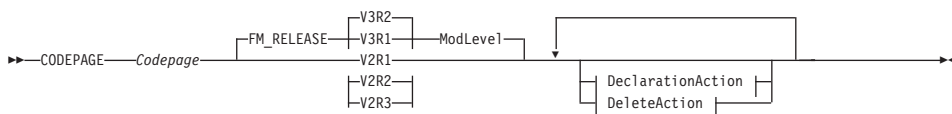
FDL ソース・ファイルの内容は、ワークフロー・オブジェクトに対して実行する任意の数の処理アクションです。有効な処理アクションには、次のものがあります。

- Create (作成)
- Replace (置換)
- Update (更新)
- Delete (削除)

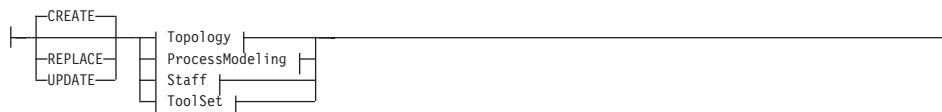
これらのアクションは、FDL 言語で指定されます。

オブジェクトを定義するために使用できる名前やストリングの構文規則については、75ページの『FDL の構文規則』を参照してください。

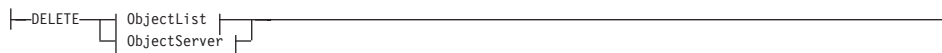
FDL ソース・ファイル



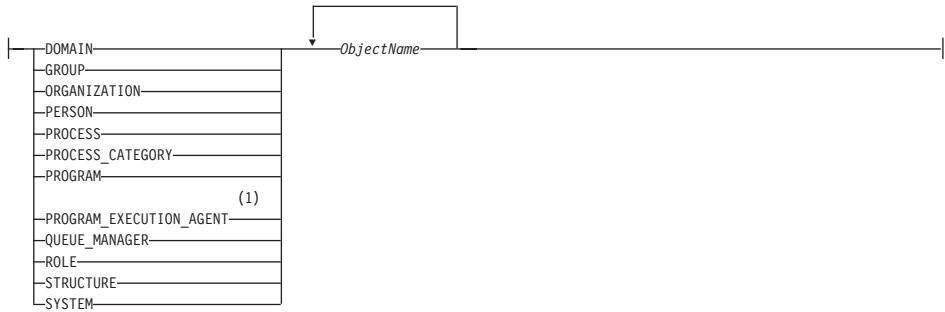
DeclarationAction:



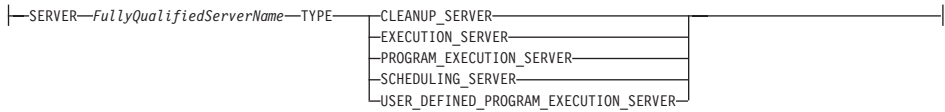
DeleteAction:



ObjectList:



ObjectServer:



注:

- 1 PROGRAM_EXECUTION_AGENT の名前は、RELATED_PERSON 属性の *PersonName* です。



処理アクション CREATE、REPLACE、UPDATE、DELETE のための、実行機能エクスポート / インポートに関するヒント

CREATE はデフォルト処理アクションです。新規のエンティティはデータベース内に作成されます。エンティティがすでに存在している場合は、システムがエラー・メッセージを出します。66ページの『搬出 / 搬入ユーティリティのオプション』に説明されているように、上書きオプション **o** は処理アクション CREATE にのみ有効です。オプション **o** を指定し、かつエンティティがデータベース内にすでに存在している場合、処理アクションは自動的に REPLACE に変更されます。

66ページの『搬出 / 搬入ユーティリティのオプション』で説明されているように、オプション **f** を使用してアクション・インポートを強制することができます。

REPLACE を使用すると、データベース内に存在する既存のエンティティすべてを置換することができます。指定したエンティティが存在していない場合、システムはエラー・メッセージを出します。もしくは、ユーザーは強制オプション **f** を指定することができます。このオプションは 66ページの『搬出 / 搬入ユーティリティのオプション』で説明されているように、**REPLACE** を **CREATE** に自動的に変更します。

UPDATE を使用する場合は、エンティティがデータベース内に存在していなければなりません。指定した属性が更新されます。強制オプション **f** およびデータベース内に存在しないオブジェクトを指定すると、66ページの『搬出 / 搬入ユーティリティのオプション』で説明されているように、**UPDATE** は自動的に **CREATE** に変更されません。単一の値に適用される場合には、その値が変更されます。反復可能項目として示されている値を複数指定した場合には、指定した値が既存の値に追加されます。たとえば、次のように指定するとします。

```
UPDATE ROLE R1 RELATED_PERSON P1 END
```

すると、人 P1 が役割に追加されます。この規則の例外は、事前定義された ROLE システム管理者には 1 人の人しか関連付けられないということです。したがって、RELATED_PERSON 属性を更新する場合には、ROLE システム管理者も置き換えることとなります。

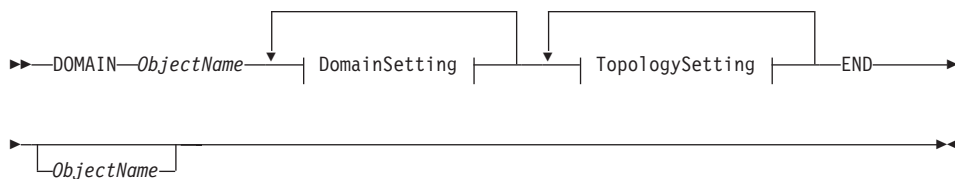
DELETE は、データベース内に存在する既存のエンティティをすべて削除するときに使用します。

定義機能搬入についての詳細およびオプションのヒントは、61ページの『定義機能搬入の開始と使用』を参照してください。

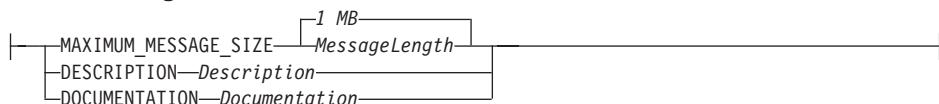
トポロジー (Topology)



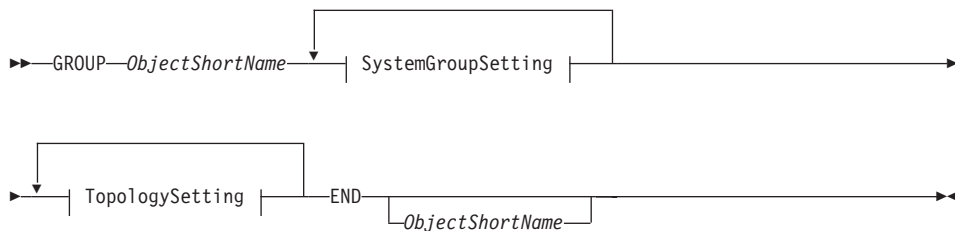
ドメイン (Domain)



DomainSetting:



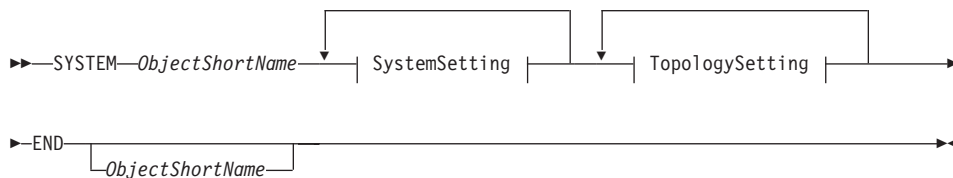
システム・グループ (System Group)



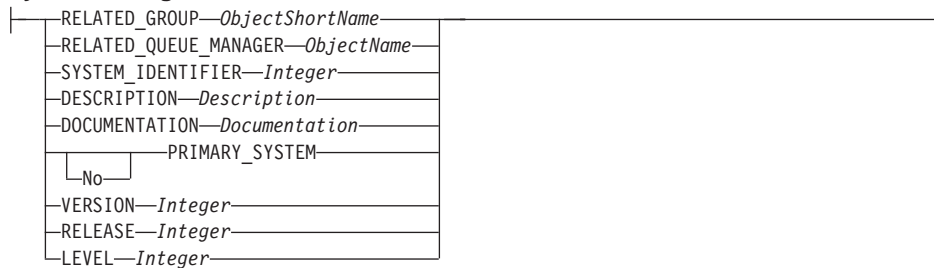
SystemGroupSetting:



システム (System)



SystemSetting:

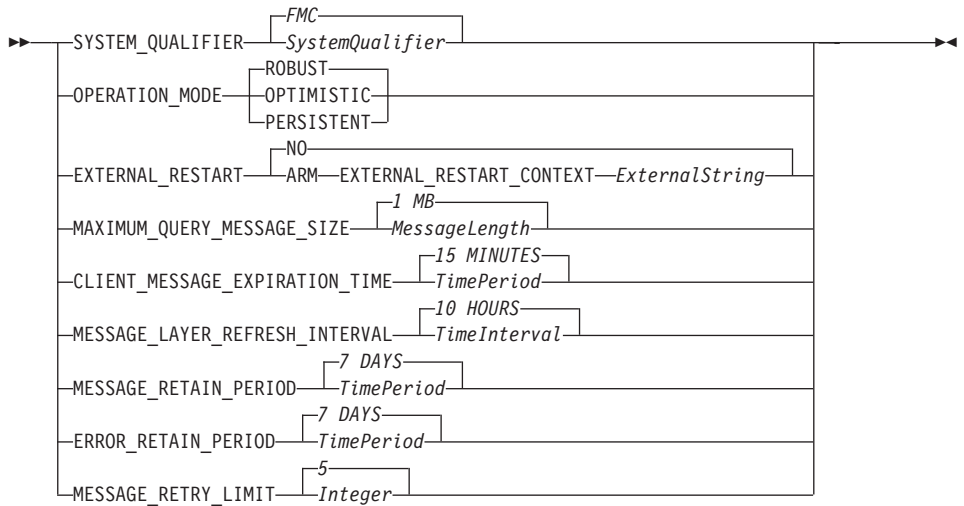


TopologySetting

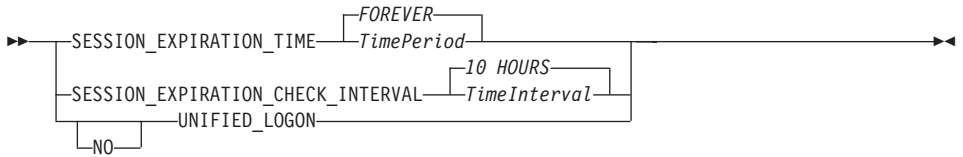
これらの属性値は必須であるため、デフォルト値はドメインにのみ適用されません。他の階層レベル (つまりシステム・グループとシステム) は、任意指定であるため、デフォルト値が設定されていません。



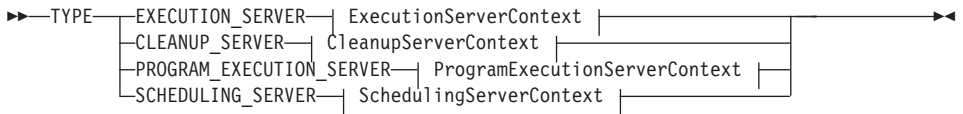
OperationSetting



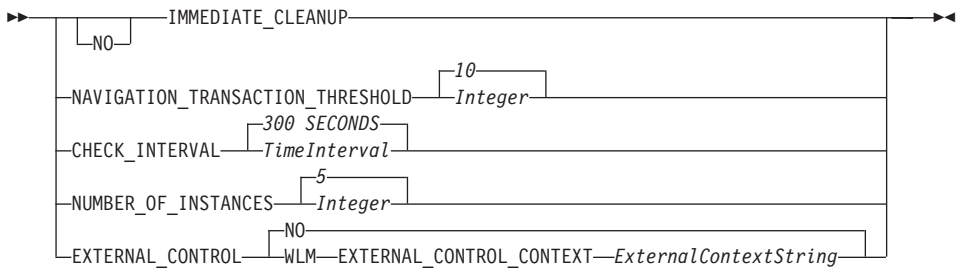
SessionSetting



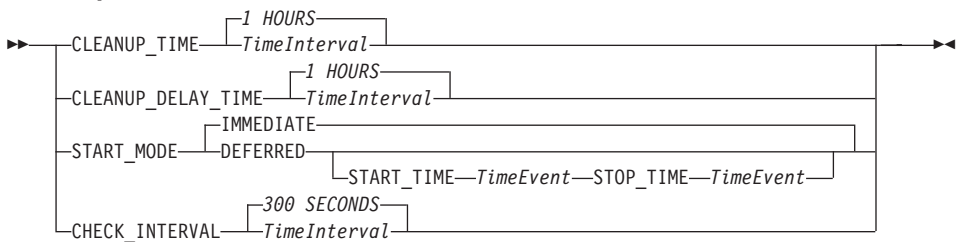
DefaultServerSetting



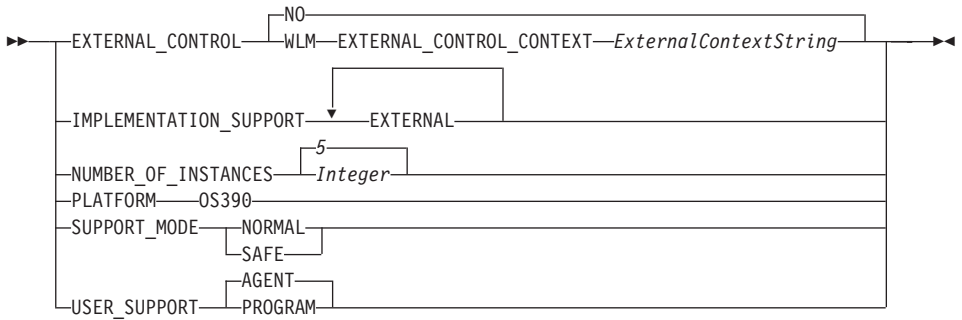
ExecutionServerContext



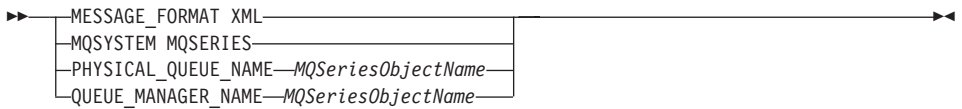
CleanupServerContext



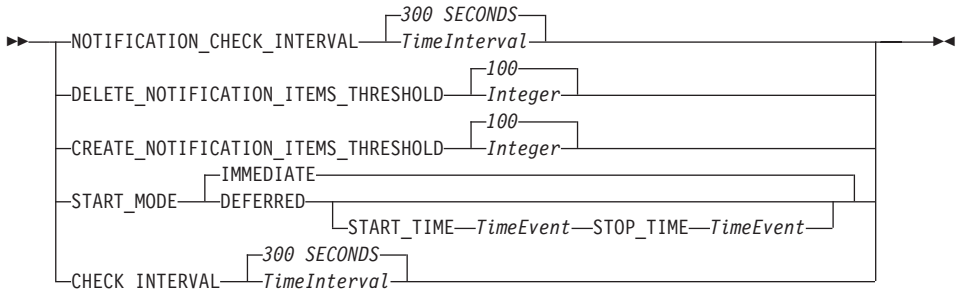
ProgramExecutionServerContext



UPESContext



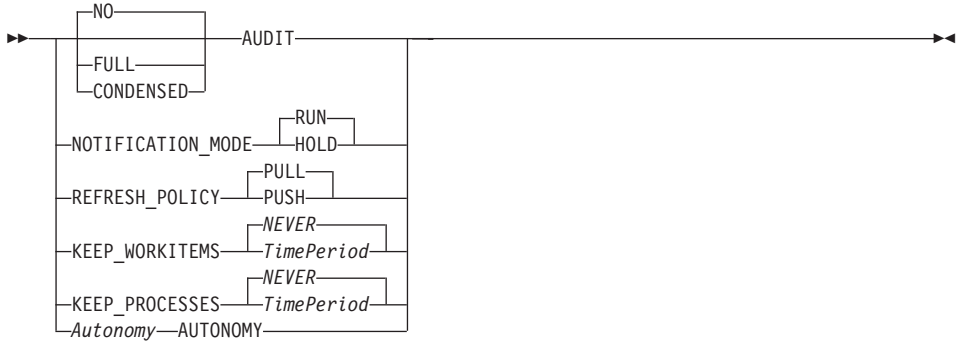
SchedulingServerContext



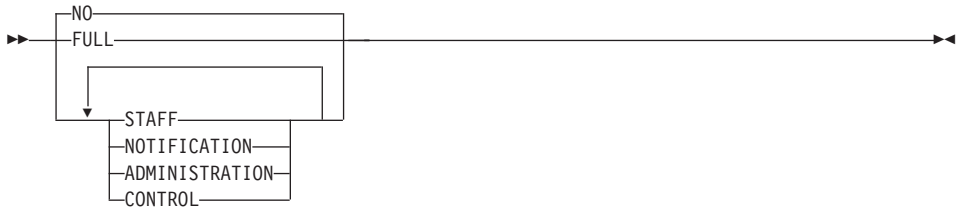
DefaultProgramExecutionAgentSetting



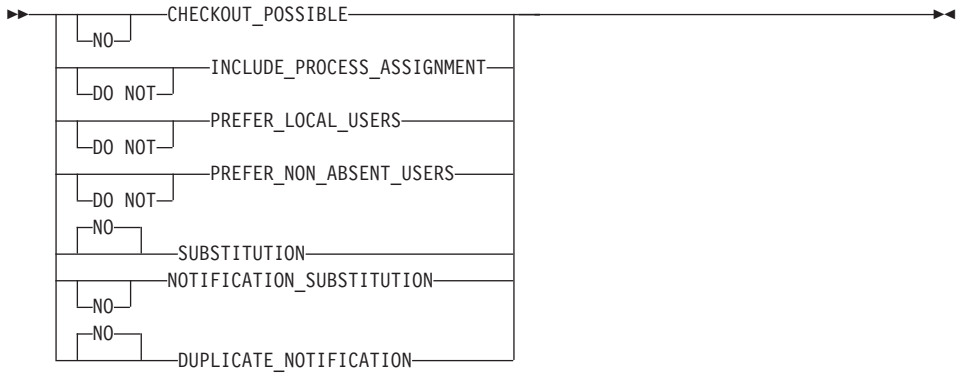
DefaultProcessSetting



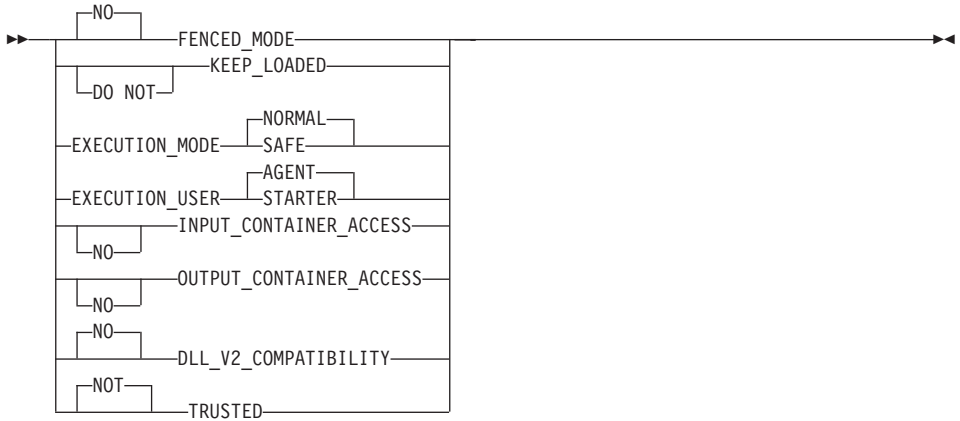
自律 (Autonomy)



DefaultActivitySetting



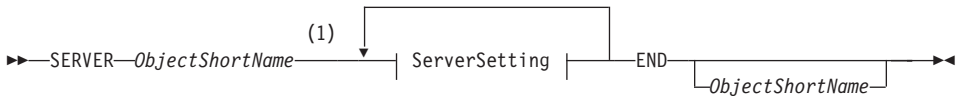
DefaultProgramSetting



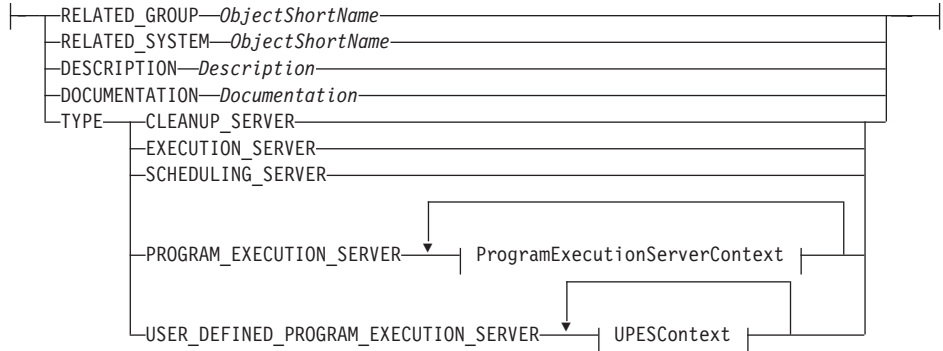
DefaultImportSetting



サーバー (Server)



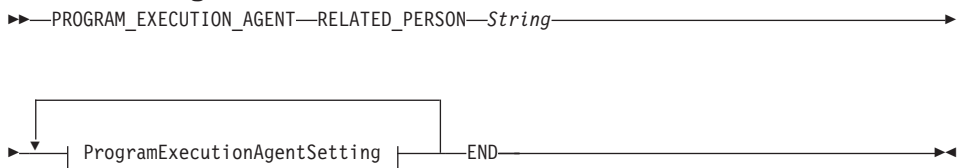
ServerSetting:



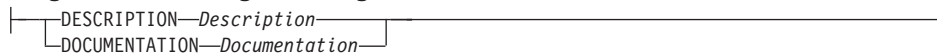
注:

- 1 サーバーの *ObjectShortName* として指定できるのは、以下のいずれかだけです。
 - EXECSVR (EXECUTION_SERVER の場合)
 - CLEANSVR (CLEANUP_SERVER の場合)
 - PESERVER (PROGRAM_EXECUTION_SERVER の場合)
 - SCHEDSVR (SCHEDULING_SERVER の場合)

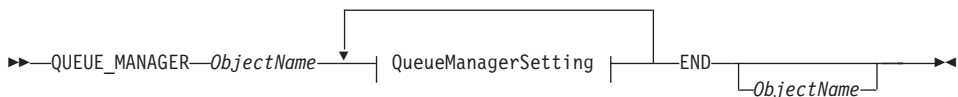
ProgramExecutionAgent



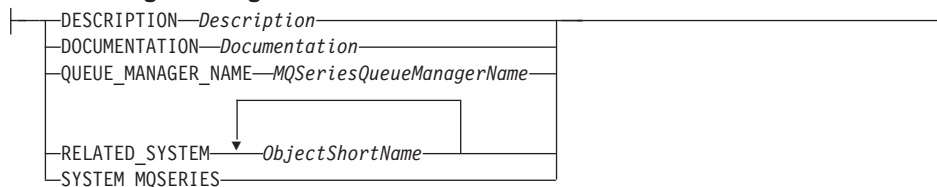
ProgramExecutionAgentSetting:



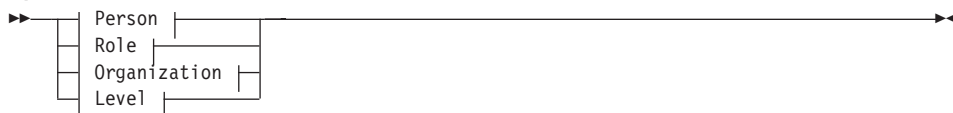
QueueManager



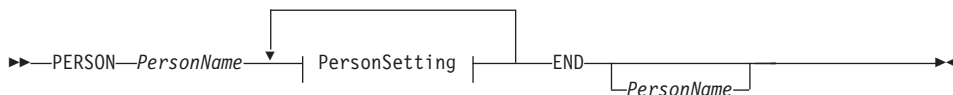
QueueManagerSetting:



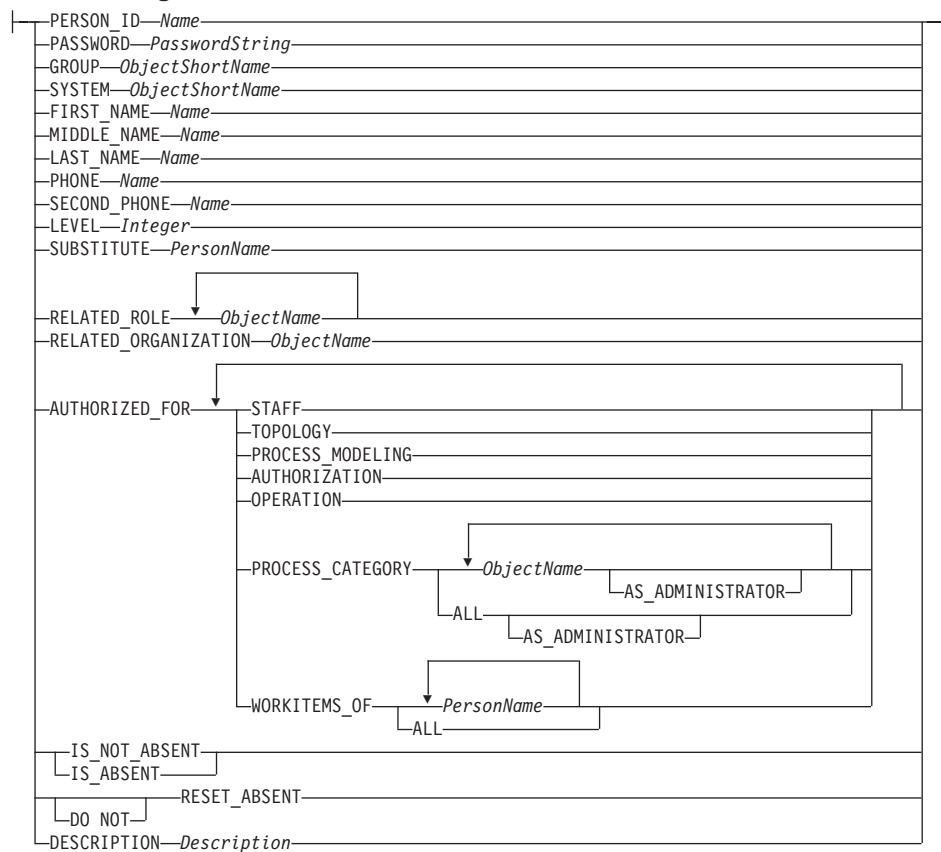
スタッフ (Staff)



人 (Person)



PersonSetting:





人 (Person) の設定のためのヒント

AUTHORIZED_FOR 属性は複数設定することができます。その場合、指定したすべての値が、該当する人に対して有効になります。ただし、**PROCESS_CATEGORY** には以下の制限が適用されます (例を参照)。

- 次のように指定できます。

```
AUTHORIZED_FOR PROCESS_CATEGORY ALL AS_ADMINISTRATOR
```

- 次の両方を指定できます。

```
AUTHORIZED_FOR PROCESS_CATEGORY ALL
AUTHORIZED_FOR PROCESS_CATEGORY 'cat1' AS_ADMINISTRATOR 'cat2' AS_ADMINISTRATOR
```

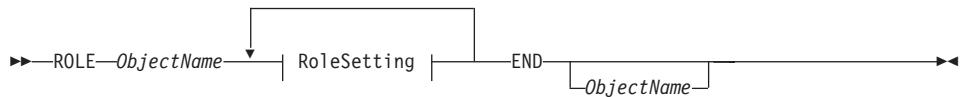
- 次のように指定できます。

```
AUTHORIZED_FOR PROCESS_CATEGORY 'cat1' AS_ADMINISTRATOR 'cat2'
```

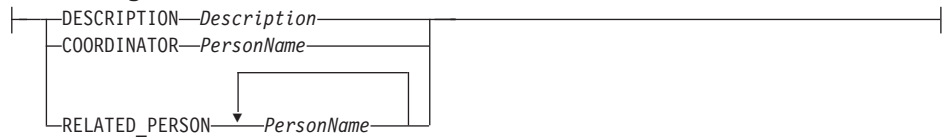
- 次の両方を指定できます。

```
AUTHORIZED_FOR PROCESS_CATEGORY 'cat1' 'cat2'
AUTHORIZED_FOR PROCESS_CATEGORY 'cat3' AS_ADMINISTRATOR
```

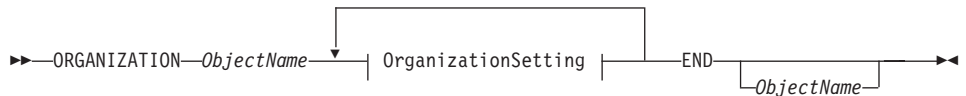
役割 (Role)



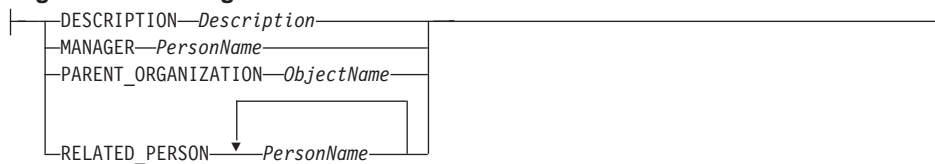
RoleSetting:



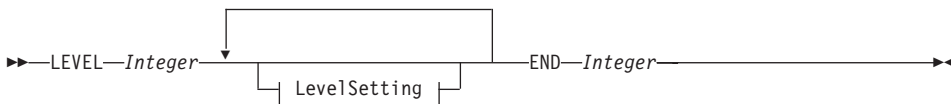
組織 (Organization)



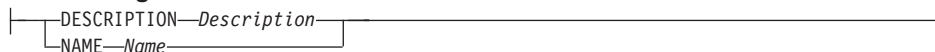
OrganizationSetting:



レベル (Level)



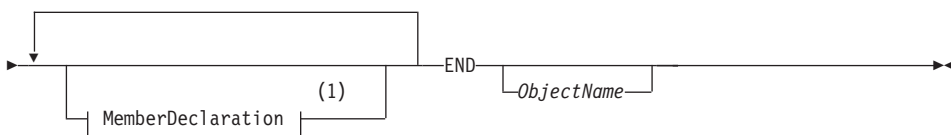
LevelSetting:



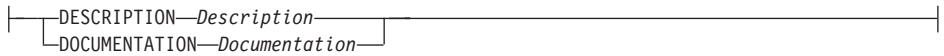
プロセス・モデル定義 (Process Modeling)



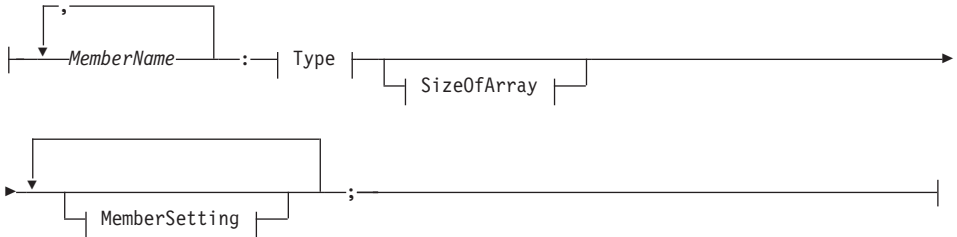
データ構造 (Data structure)



StructureSetting:



MemberDeclaration:



Type:



SizeOfArray:



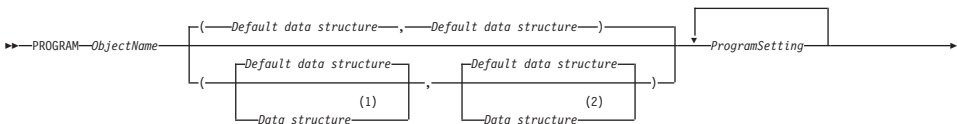
MemberSetting:

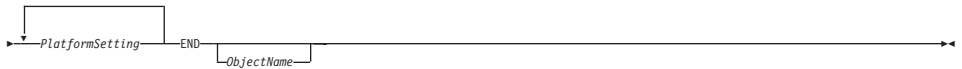


注:

- 1 構造内のメンバーは 512 以下です。
- 2 指定できる最大サイズは 512 要素です。

プログラム (Program)

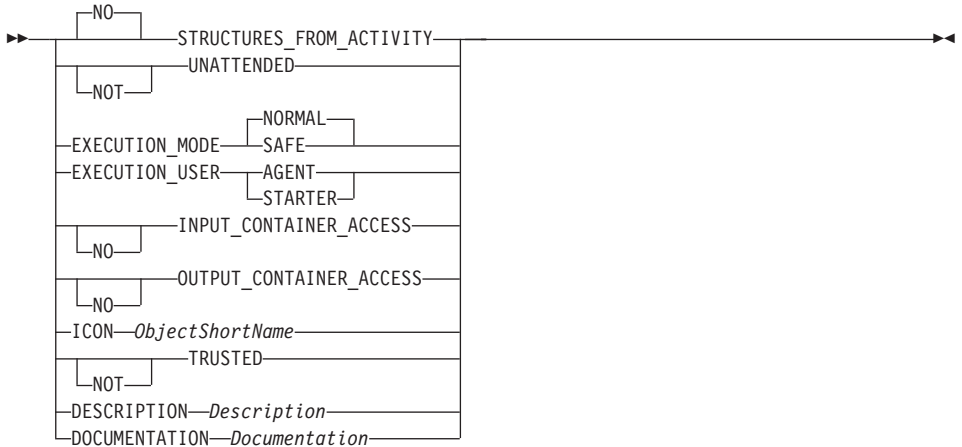




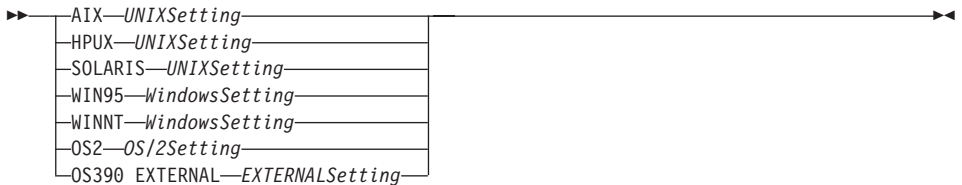
注:

- 1 指定する最初のデータ構造は入力データ構造です。
- 2 指定する第 2 のデータ構造は出力データ構造です。

ProgramSetting



PlatformSetting



UNIXSetting



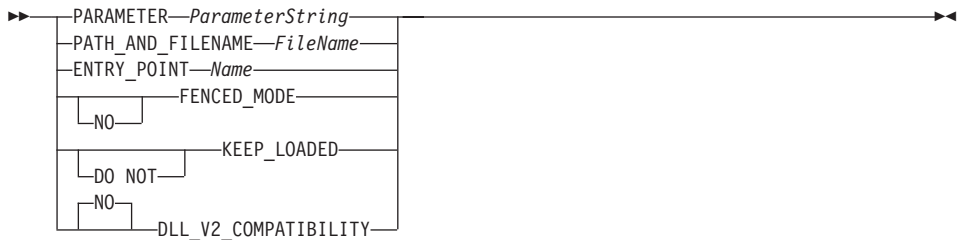
WindowsSetting



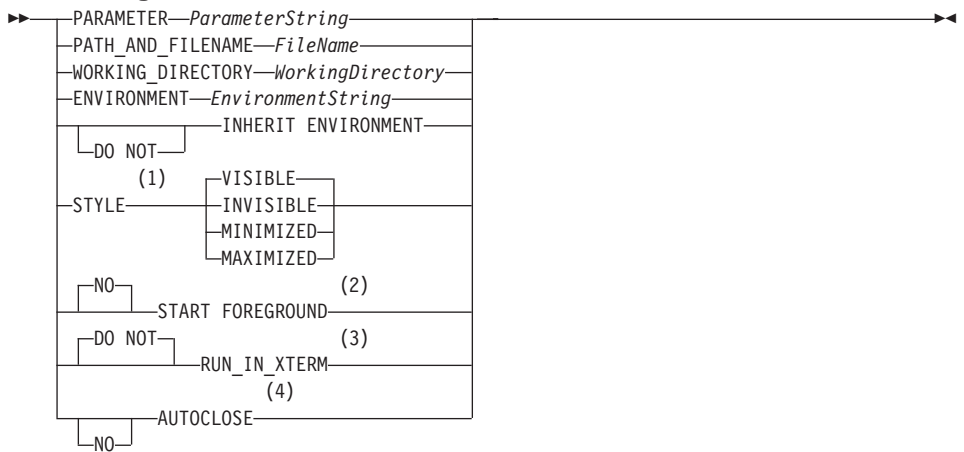
OS/2Setting



DLLSetting



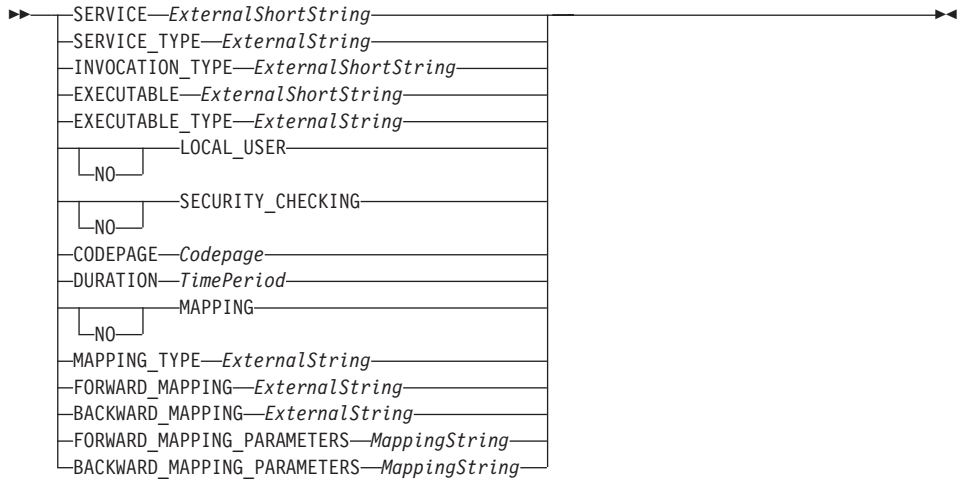
EXESetting



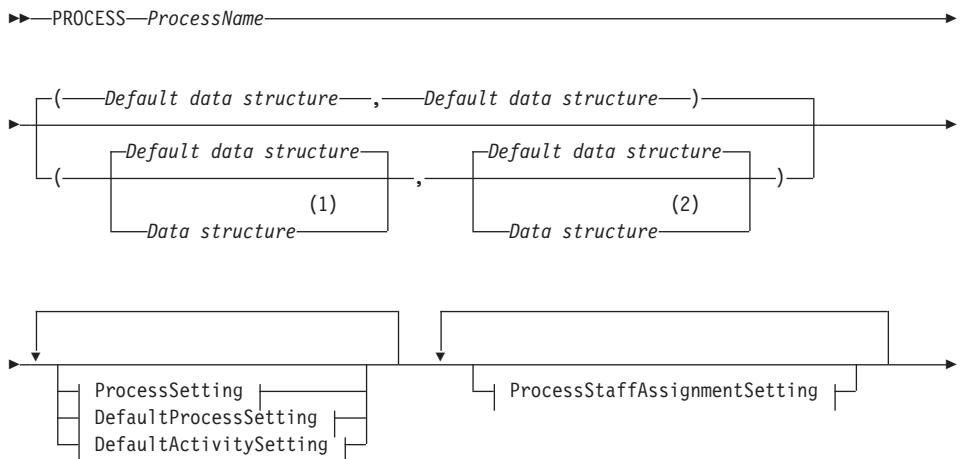
注:

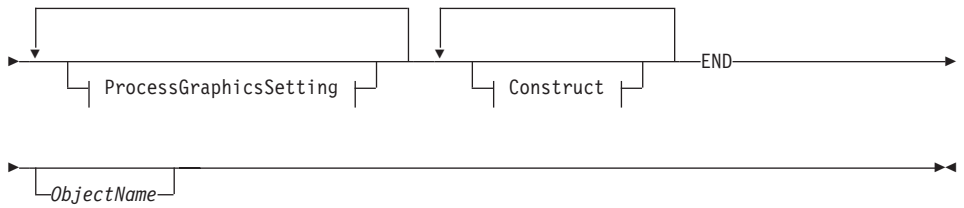
- 1 Windows 9x、Windows NT、および OS/2 の場合のみ
- 2 Windows 9x、Windows NT、および OS/2 の場合のみ
- 3 UNIX のみ
- 4 OS/2 のみ

EXTERNALSetting



プロセス (Process)

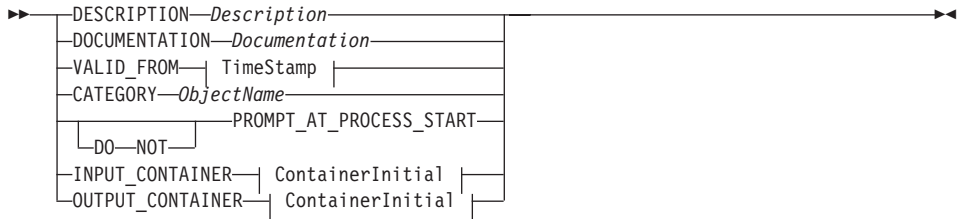




注:

- 1 指定する最初のデータ構造は入力データ構造です。
- 2 指定する第 2 のデータ構造は出力データ構造です。

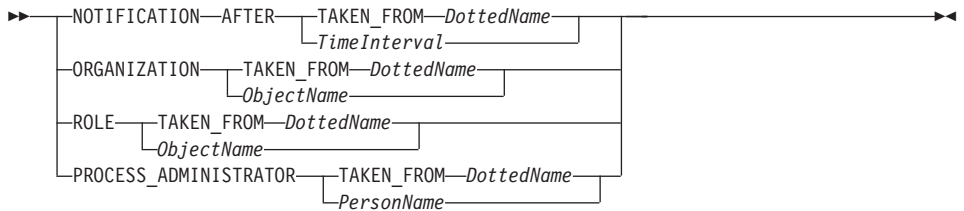
ProcessSetting



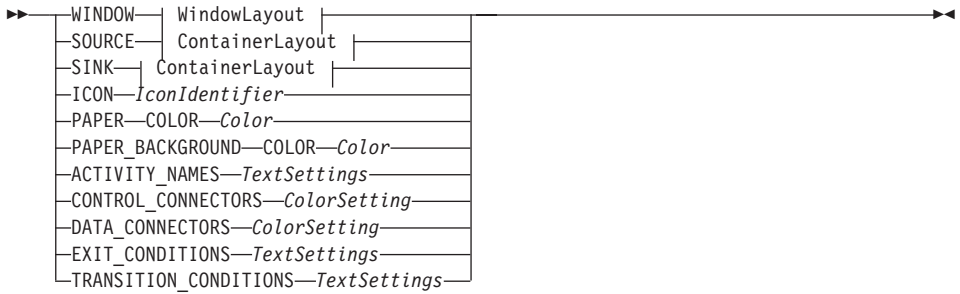
ProcessStaffAssignmentSetting



ExplicitProcessStaffAssignmentSetting



ProcessGraphicsSetting



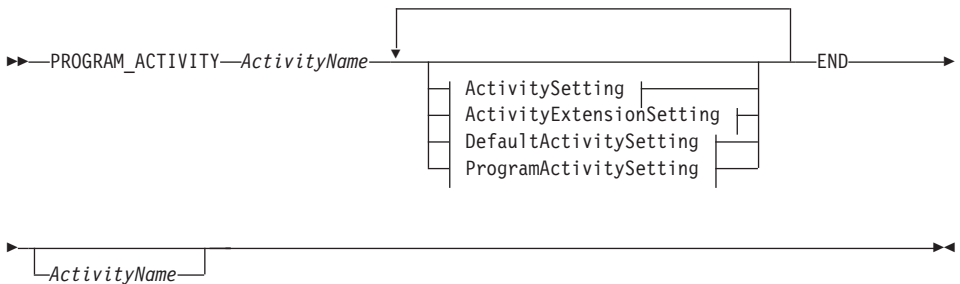
構成 (Construct)



アクティビティー (Activity)

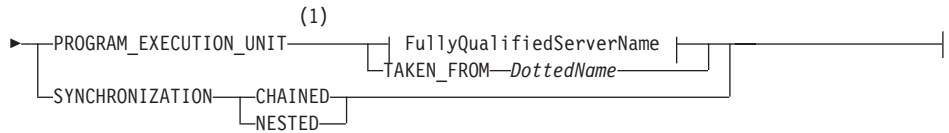


ProgramActivity



ProgramActivitySetting:

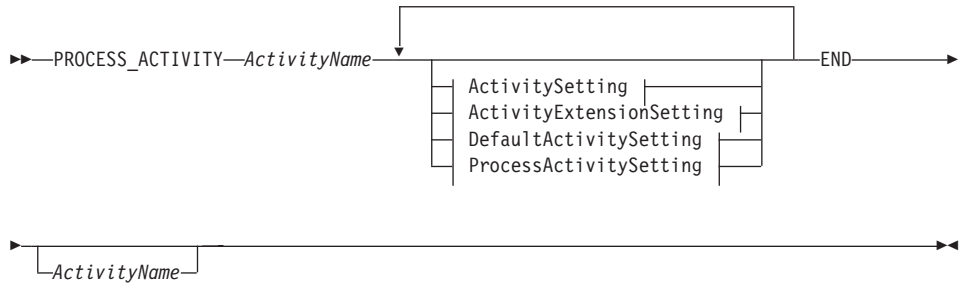




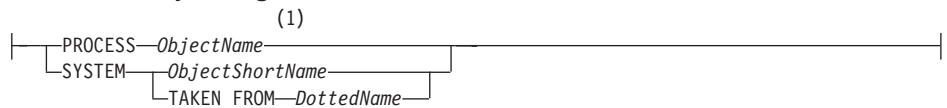
注:

- 1 PROGRAM_EXECUTION_UNIT の代わりにキーワード PROGRAM_EXECUTION_SERVER もまだ使用できます。しかし、新しい定義の場合は、このリリースでは旧キーワードは一時的な解決としてのみ有効なので、PROGRAM_EXECUTION_UNIT だけを使用するようにしてください。サーバーの種類は PROGRAM_EXECUTION_SERVER または USER_DEFINED_PROGRAM_EXECUTION_SERVER です。

ProcessActivity



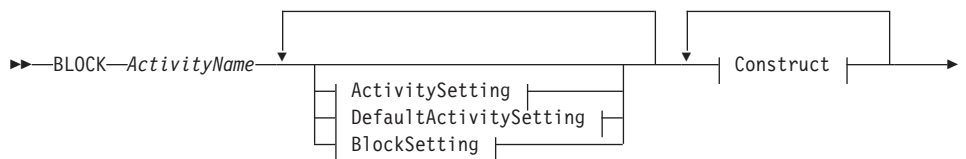
ProcessActivitySetting:

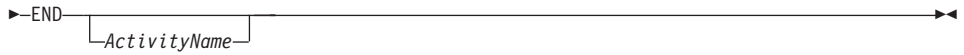


注:

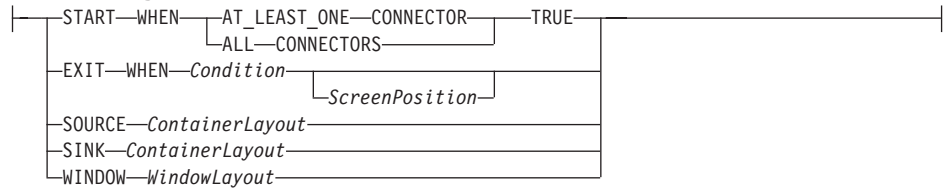
- 1 これはプロセスの名前です。

ブロック (Block)

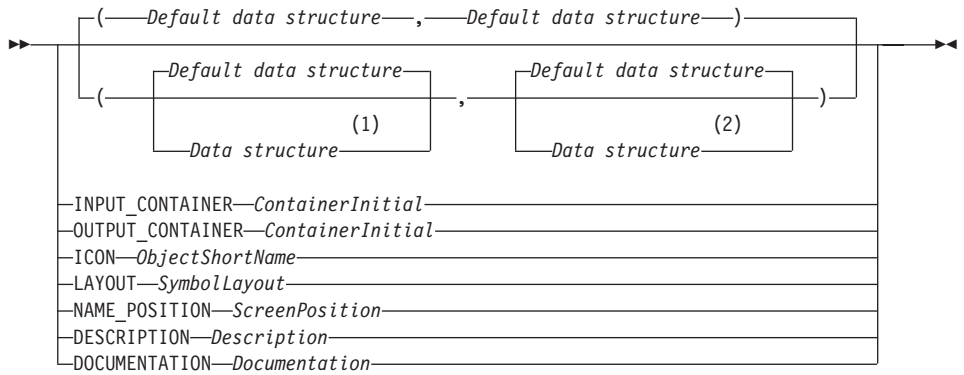




BlockSetting:



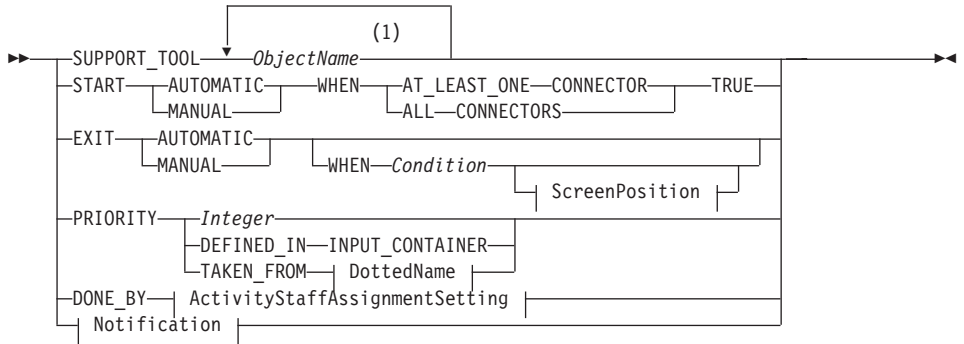
ActivitySetting



注:

- 1 指定する最初のデータ構造は入力データ構造です。
- 2 指定する第 2 のデータ構造は出力データ構造です。

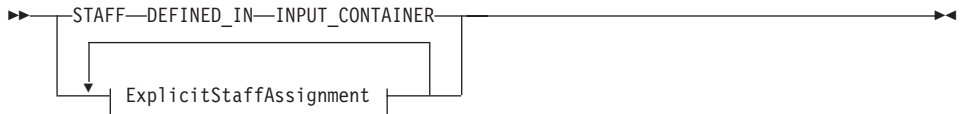
ActivityExtensionSetting



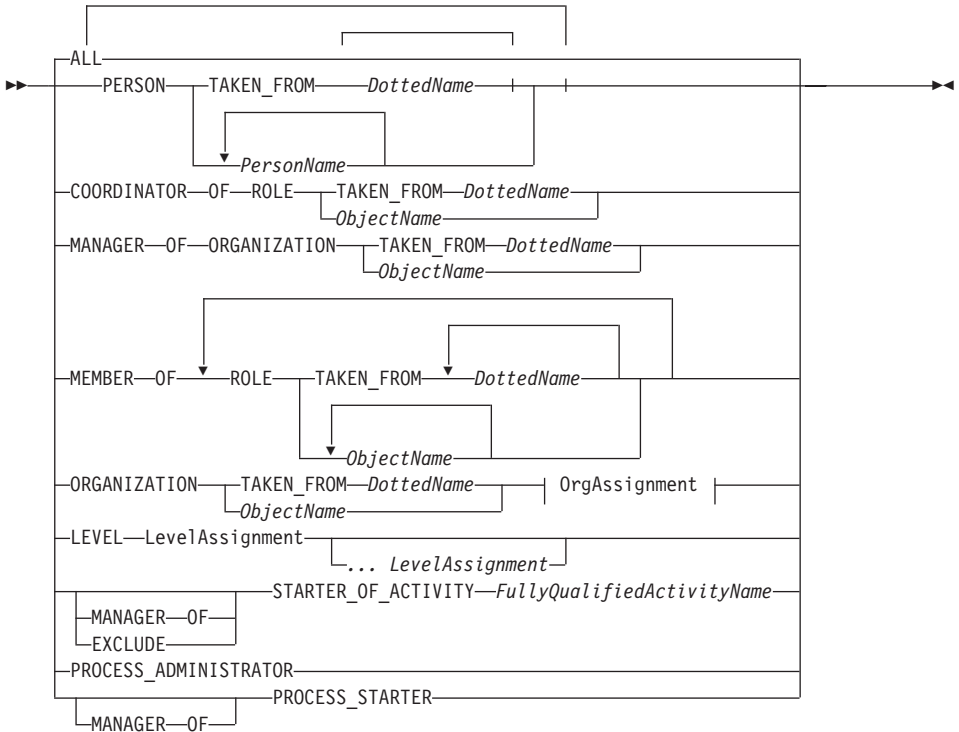
注:

- 1 これはプログラムの名前です。

ActivityStaffAssignmentSetting



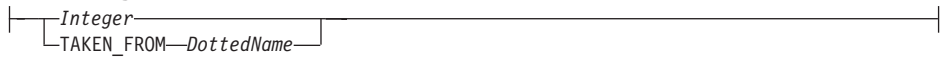
ExplicitStaffAssignment:



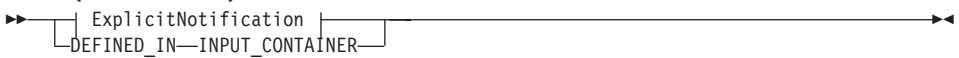
OrgAssignment:



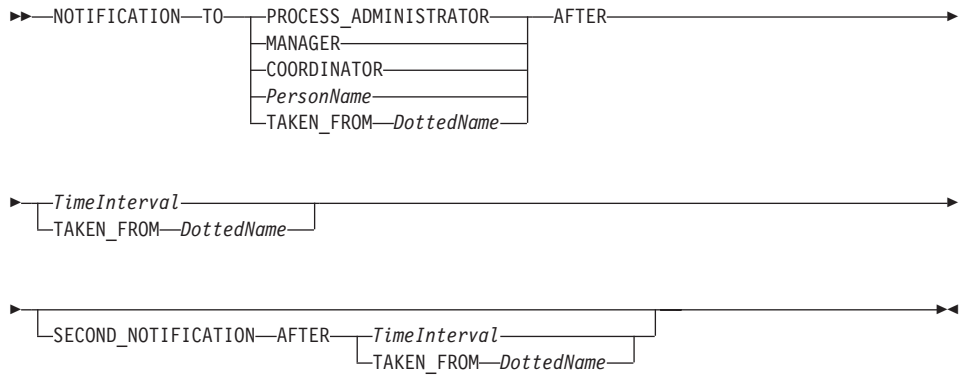
LevelAssignment:



通知 (Notification)



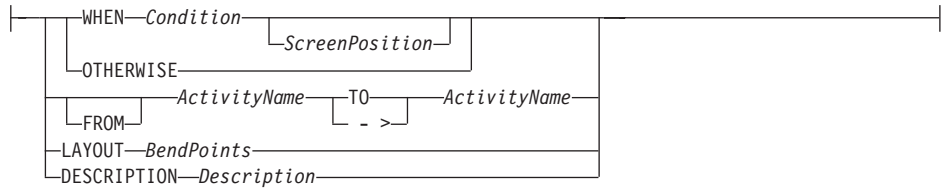
ExplicitNotification:



ControlFlow



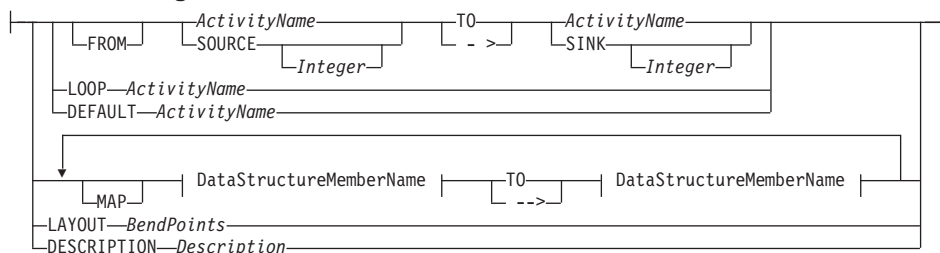
ControlSetting:



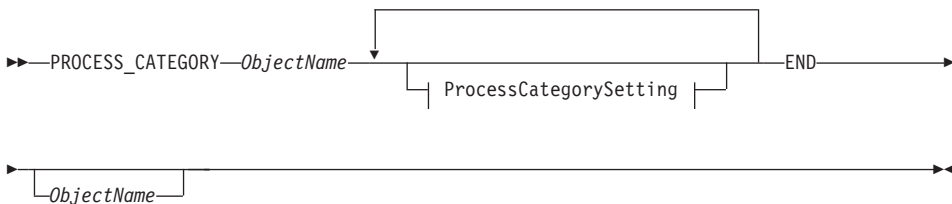
DataFlow



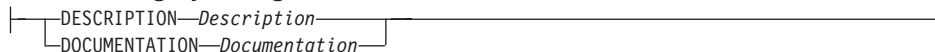
DataflowSetting:



プロセス・カテゴリー (Process category)



ProcessCategorySetting:

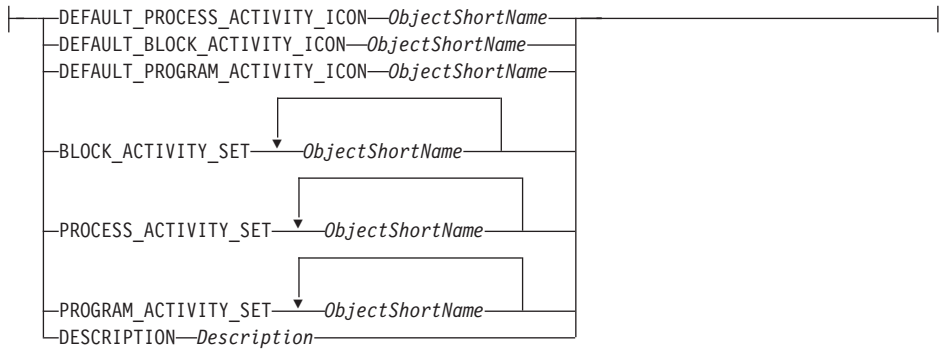


ツール・セット (ToolSet)

ツール・セットは、MQSeries Workflow 定義機能の場合のみ有効です。



ToolSetSetting:



以下のデフォルト値が適用されます。

```

TOOL_SET 'STANDARD'
  DEFAULT_PROCESS_ACTIVITY_ICON 'fmcblrca'
  DEFAULT_BLOCK_ACTIVITY_ICON 'fmcblka'
  DEFAULT_PROGRAM_ACTIVITY_ICON 'fmcblrpa'
END 'STANDARD'
  
```

共通変数

ScreenPosition

(1)
 ▶▶—XPOS—Integer—YPOS—Integer—▶▶

注:

1 Integer は 0.1 mm を表します。

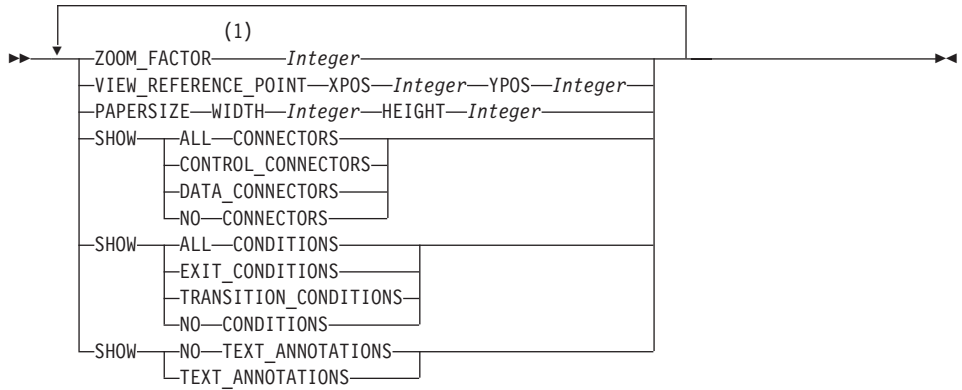
SymbolLayout

▶▶—ScreenPosition—[WIDTH—Integer—HEIGHT—Integer]—▶▶

ContainerLayout

▶▶—Integer—| SymbolLayout |—▶▶

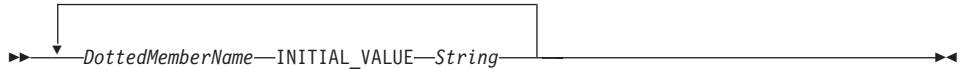
WindowLayout



注:

- 1 指定できる範囲は 10~200 です。

ContainerInitial



BendPoints



カラー (Color)



RedPart:



GreenPart:





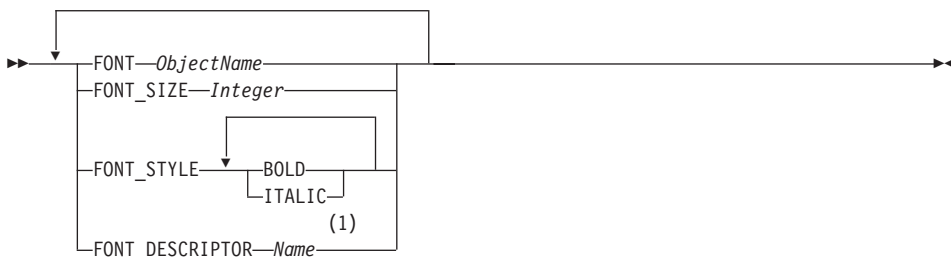
ColorSetting



TextSettings



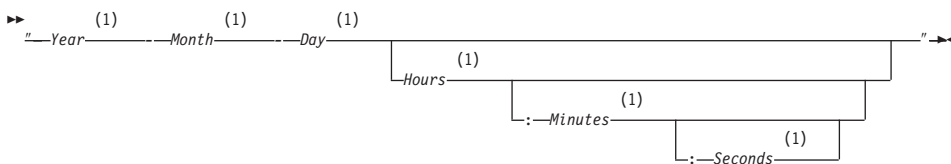
FontSettings



注:

- 1 FONT_DESCRIPTOR はプラットフォーム固有の設定値で、Windows NT/9x で使用する文字セットなどの追加情報が入ります。

TimeStamp



注:

- 1 UTC 時間を指定する整数。例: 1999-06-18 12:29:05

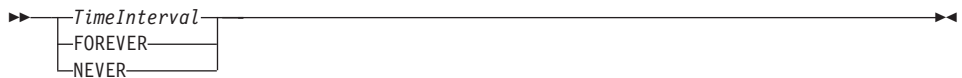
TimeInterval



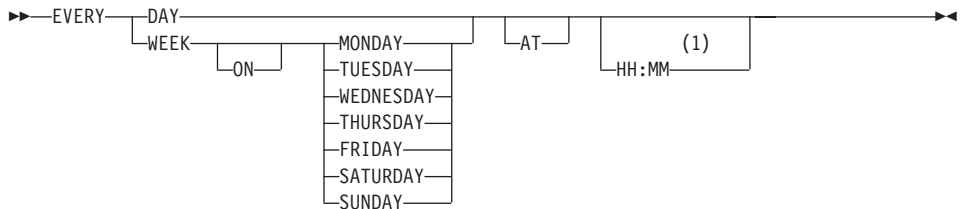
注:

- 1 YEARS および MONTHS を使用できるのは、107ページの『ProcessStaffAssignmentSetting』の Explicit Process Staff Assignment Setting の NOTIFICATION AFTER の場合と、112ページの『通知 (Notification)』の Explicit Notification TO の場合だけです。

TimePeriod



TimeEvent



注:

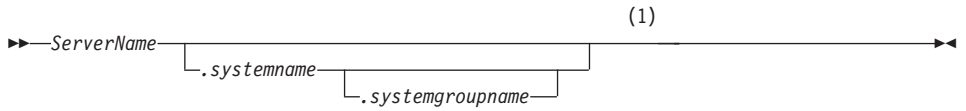
- 1 現地時間を指定する整数。

MessageLength

最小長は 256 KB、最大長は 96 MB です。



FullyQualifiedServerName



注:

- 1 ServerName、systemname、systemgroupname は、オブジェクト短縮名です。systemname と systemgroupname の省略時値はプロファイルから取られます。それは、インストール時に生成されるものです。

第3部 付録および後付け

付録A. スタッフ定義のモデル化についての詳細

定義機能のワーク・フロー・モデルを定義するときには、3ページの『ワークフロー・モデルとは ?』で説明されているように、アクティビティー、人員の編成、および IT リソースと共にビジネス・プロセスを定義します。この定義は、実行機能のユーザーがプロセスのインスタンスを開始するときにいつでも適用されます。これらの定義には、実行すべきアクティビティーのスタッフ割り当てが含まれます。詳細については、39ページの『第4章 スタッフの割り当てとプロセス・フローの定義』を参照してください。この節では、実行時に柔軟で確実なスタッフ解決を行うためのモデル化について説明します。

ワーク・フロー・モデル内の定義を実行時に使用できるようにする前に、それらの定義がどのように検査されるのかを知りたい場合は、51ページの『ワークフロー・モデルの検証』を参照してください。

スタッフ解決とは ?

プロセス・インスタンスが実行機能で実行されるときや、プログラムまたはプロセス・アクティビティーが開始するときには、定義機能でモデル化されているスタッフ定義が実行時に解決されます。作業項目が作成され、すべての適格な実行機能ユーザーのワーク・リストに表示されます。一般的な規則では、ユーザーはワーク・リストに表示された作業項目だけを開始することができます。

MQSeries Workflow では、スタッフ解決に影響する多くのオプションを指定できます。これにより、ユーザーは柔軟性の高いワーク・フロー・モデルを定義できます。以下の節では、指定できるオプションの概要を示します。

アクティビティーのためのスタッフ解決の定義

定義機能にプロセスを定義する際に、アクティビティーのためのスタッフ定義を指定することができます。39ページの『アクティビティーのプロパティーの指定』にはその方法が表示されています。「**スタッフ 1 (Staff 1)**」および「**スタッフ 2 (Staff 2)**」のページでは、スタッフ解決に使用される方式を選択することができます。次のことが可能です。

- 124ページの『アクティビティーのプロパティーを基にしたスタッフ解決』に説明されているように、「**スタッフ 1 (Staff 1)**」および「**スタッフ 2 (Staff 2)**」のプロパティー・ページで、スタッフ解決の基準を定義します。

- 127ページの『コンテナ・メンバーを基にしたスタッフ解決』に説明されているように、実行時にスタッフ解決が入力コンテナから取られるように定義します。

アクティビティーのプロパティーを基にしたスタッフ解決

アクティビティーのためのスタッフ解決を定義する場合、「**スタッフ 1 (Staff 1)**」ページのオプションを選択することができます。アクティビティーは定義された基準を満たした人に割り当てられます。それには次のようなものがあります。

- **すべての人 (All people)**
- **定義済みメンバーのスタッフ (Staff from predefined members)**。 127ページの『コンテナ・メンバーを基にしたスタッフ解決』で説明されています。
- **役割コーディネーター (Coordinator of role)**
- **組織マネージャー (Manager of organization)**
- **人 (People)**
- **プロセス管理者 (process administrator)**
- **プロセス開始者 (Process starter)**
- **プロセス開始者のマネージャー (Manager of process starter)**
- **アクティビティーの開始者 (Starter of activity)**
- **アクティビティーの開始者のマネージャー (Manager of starter of activity)**
- **アクティビティーの開始者の除外 (Exclude starter of activity)**

オプション「**アクティビティーの開始者の除外 (Exclude starter of activity)**」を選択する場合、以下の事柄を考慮する必要があります。

- 2つのアクティビティー、*A1* と *A2* があると仮定します。アクティビティー *A1* のスタッフ解決は役割に基づいています。アクティビティー *A2* のスタッフ解決では、*A1* を対象に、「**アクティビティーの開始者の除外 (Exclude starter of activity)**」が定義されています。プロセス・モデルで、*A1* は *A2* の前に実行されます。すると、アクティビティー *A2* のスタッフ解決は、次のようになります。

A1 に定義されたスタッフ解決が *A2* で再び行われ、アクティビティー *A1* の開始者が除外されます。

オプション「**アクティビティーの開始者の除外 (Exclude starter of activity)**」は、検査および平衡をとるために使用することができます。この

場合、同じグループの人々は、A1 および A2 の両方のアクティビティーに対して適格です。ただし、A1 および A2 を同じ人が開始することはできないという制約があります。

- このオプションは、たとえば、3 つ以上のアクティビティーを異なる人によって実行されるようにしたい場合にも選択できます。

A1、A2、 および A3 という 3 つのアクティビティーがあると仮定します。これらは次のような順序で実行されます。

- アクティビティー A1 のスタッフ解決は役割に基づいています。
- アクティビティー A2 のスタッフ解決は、A1 を対象に「**アクティビティーの開始者の除外 (Exclude starter of activity)**」が設定されています。
- アクティビティー A3 のスタッフ解決は、A2 を対象に「**アクティビティーの開始者の除外 (Exclude starter of activity)**」が設定されています。

すると、アクティビティー A1 および A2 のスタッフ解決は、前述の通りに実行されます。アクティビティー A3 のスタッフ解決は、以下のようになります。

A1 に定義されたスタッフ解決が A3 で再び行われ、その結果としてアクティビティー A1 および A2 の開始者が除外されます。この例は、4 つ以上のアクティビティーについても適用できます。



定義されている制御パスを基準にして、現在のアクティビティーに先行しているアクティビティーだけを参照できます。

- 以下の例は、このオプションを使用するときには考慮する必要のある事柄を示しています。

5 つのアクティビティー (A1 ~ A5) が含まれている、次のようなプロセスがあるとします。

A1 → A2 → A3 → A4 → A5

アクティビティー A2 については、次のような定義を行うことができます。

- A1 を対象にしたアクティビティーの開始者の除外は定義できません。
- A4 または A3 を対象にしたアクティビティーの開始者の除外は定義できません。

「**スタッフ 2 (Staff 2)**」ページでは、動的スタッフ割り当てを定義できます。これは、39ページの『アクティビティーへのスタッフの割り当て』、40ページの『動的スタッフ割り当ての指定』、およびオンライン・ヘルプで説明されています。

役割、組織、およびレベルを基にしたスタッフ解決には、フィルター基準を指定することができます。フィルター基準を指定すると、実行時に、すべてのフィルター基準に当てはまる人だけが作業項目を得ることができるようになります。

役割メンバー (Members of roles)

実行時に作業項目を得るよう指定されているすべての役割のメンバーでなければなりません。

組織 (Organization)

次のように定義できます。

- 組織 (Organization)
- コンテナにある (From container)

さらに適用できる基準を指定することができます。それには次のものが含まれます。

- メンバーのみ (Members only)
- 上位管理者 (Reporting managers)
- 下位組織 (Child organizations)

レベル (Level)

次のように定義できます。

- 下限レベル (From Level)
- 上限レベル (To Level)

これで、フィルター基準の間隔を指定することができます。

アクティビティー・プロパティー・ウィンドウの「**制御 (Control)**」ページには、他にもスタッフ解決のためのオプションがあります。これについてはオンライン・ヘルプで説明されています。

以下のオプションはスタッフ解決の動作に直接影響します。

- プロセス・レベルで定義できるスタッフ解決設定があります。「**プロセス割り当ての組み込み (Include process assignment)**」が選択された場合、これらの設定がアクティビティーのスタッフ解決に組み込まれます。「**継承 (Inherited)**」を指定する場合の考慮事項についての詳細は、130ページの『継承するスタッフ解決の定義』を参照してください。
- 継承の必要がない場合は、それぞれのオプションごとに、「**継承 (Inherited)**」を選択しないように切り替えておくこともできます。詳細については、129ページの『プロセス・レベルでのスタッフ解決の定義』を参照してください。

- 「ローカル・ユーザーの優先 (Prefer local users)」は実行機能ではまだ使用できませんが、将来実装される可能性があります。
- 「非不在ユーザーの優先 (Prefer not absent users)」は、不在として定義されていないユーザーだけを考慮しようとしていることを指定します。ただし、適格ユーザーがすべて不在の場合は、すべてのユーザーが有効なユーザーであると見なされます。
- 「ユーザーが不在の場合の代理人割り当て (Assign substitute if user is absent)」は、不在として定義されていないユーザーのみを適格ユーザーとして指定します。代理人は、不在として宣言されていない場合にのみ適格ユーザーと見なされます。適格ユーザーがいない場合、作業項目はプロセス管理者に割り当てられます。このオプションが設定されると、「非不在ユーザーの優先 (Prefer not absent users)」の値は無視されます。



アクティビティのスタッフ解決中に適格ユーザーが見つからない場合、アクティビティはプロセス管理者に割り当てられます。

コンテナ・メンバーを基にしたスタッフ解決

ワークフロー・モデルでは、プロセスの入出力データ、そのアクティビティ、およびその中のブロックに記憶域が割り振られます。どのアクティビティにも、入力用のデータ・コンテナと出力用のデータ・コンテナがあります。各データ・コンテナはデータ構造によって定義されます。定義済みデータ・メンバーを含むデータ構造についての詳細は、*IBM MQSeries Workflow: プログラミングの手引き* を参照してください。

ユーザーを明確に定義せずに、コンテナ・メンバーを基にしたスタッフ解決を定義することもできます。コンテナ・メンバーを基にしたアクティビティのためのスタッフ解決を定義する場合は、「**スタッフ 1 (Staff 1)**」ページ、または「**スタッフ 2 (Staff 2)**」ページのオプションを選択することができます。アクティビティは定義された基準を満たした人に割り当てられます。

コンテナ・メンバーを基にしたスタッフ解決では、実行時に解決されるアクティビティのプロパティを動的に指定することができます。次のことが可能です。

- 定義済みコンテナ・メンバーからスタッフを割り当てます。
- 1 つまたは複数のコンテナ・メンバーを指定します。

定義済みコンテナ・メンバーの使用

定義済みのコンテナ・メンバーを使用する場合、次の順序で処理されます。

1. `_ACTIVITY_INFO.People` がセットされている場合、指定された人のリストに作業項目が割り当てられます。他のコンテナ・メンバーの内容はすべて無視されます。
2. `_ACTIVITY_INFO.CoordinatorOfRole` がセットされている場合、指定された役割のコーディネーターに作業項目が割り当てられます。他のメンバーの内容はすべて無視されます。
3. 次のメンバーは並行して評価することができます。
 - `_ACTIVITY_INFO.MembersOfRoles`
 - `_ACTIVITY_INFO.Organization`
 - `_ACTIVITY_INFO.OrganizationType`
 - `_ACTIVITY_INFO.LowerLevel`
 - `_ACTIVITY_INFO.UpperLevel`

これらのフィルター基準をすべて満たす人だけが、作業項目を実行時に受け取ることができます。たとえば、2 つの役割 (1 つの組織、最低レベルが 4) が指定されている場合は、2 つの役割のメンバーで、この組織のメンバーであると同時に、最低 4 に定義されたレベルを持つ人だけが作業項目を受信できます。

`_ACTIVITY_INFO.Organization` が設定されている場合、メンバー `_ACTIVITY_INFO.OrganizationType` は次のような意味を持ちます。

- このメンバーが 0 に設定されているか、全く設定されていない場合、この組織とその下位組織のメンバーはすべて適格です。
- このメンバーが 1 に設定されている場合、この組織のメンバーと、その直接の下位組織のマネージャーはすべて適格です。

既存のアプリケーションとの互換性のため、

`_ACTIVITY_INFO.OrganizationType` の 0 以外の値は 1 と解釈されます。

コンテナ・メンバーの指定

「スタッフ 1 (Staff 1)」および「スタッフ 2 (Staff 2)」ページにある特定のスタッフ解決オプションでは、複数のコンテナ・メンバーを明確に定義することができます。複数のユーザーまたは役割は、コンテナ・メンバーの中では、セミコロン (;) で分けられます。



人または 役割のみを指定することはできますが、人と役割を混合することはできません。

コンテナ・メンバーを基にしたスタッフ解決は、MQSeries Workflow 外のスタッフ解決の実行にも使用できます。次のようなシナリオが考えられます。

1. プログラム・アクティビティーが開始され、その結果、幾人かが解決し、出力コンテナに格納されました。
2. このコンテナ・フィールドは、他のアクティビティーの入力コンテナ・フィールドにマップされています。
3. 最終的に、このコンテナ・フィールドに格納されたすべての人がこのアクティビティーの作業項目を受信しました。

プロセス・レベルでのスタッフ解決の定義

プロセスに有効なスタッフ定義を指定するには、次のように行います。

1. ツリー・ビューで、プロパティーを定義したいプログラム・アクティビティーを右クリックします。
2. 「プロパティー (Properties)」をクリックします。プロセスのプロパティー・ウィンドウがオープンします。
3. 「スタッフ (Staff)」ページをクリックします。

「スタッフ (Staff)」では、次のうちのいずれかを指定できます。

定義済みメンバーのスタッフ

`_PROCESS_INFO.Role` および `_PROCESS_INFO.Organization` データ・メンバーを使用して、1つの役割、または1つの組織を指定することができます。実行時にプロセス・インスタンスを開始する場合、プロセス管理者は次のように決定されます。

- `_PROCESS_INFO.ProcessAdministrator` が設定されている場合、プロセス管理者を決定するための値がとられます。
- このメンバーが設定されておらず、プロセスがサブプロセスであってもスタッフ自律が定義されていない場合、親プロセスのプロセス管理者がプロセス管理者として定義されます。
- その他のケースでは、プロセス開始者がプロセス管理者として定義されます。
- ただし、プロセス開始者がいない場合は、システム管理者がプロセス管理者として定義されます。

または

プロセス管理者、役割、組織

プロセス管理者、つまり人 (**Person**) を明確に指定するか、「**コンテナにある (From container)**」を選択してプロセス入力コンテナからとられるようにします。

プロセス管理者には、プロセス・インスタンスを正常に開始して終了できるようにする責任があります。たとえば、スタッフ解決の間に適格ユーザーが割り当てられない場合、作業項目はプロセス管理者に割り当てられます。

アクティビティーのプロセス定義の使用

アクティビティーが属するプロセスに定義されているとおりに、アクティビティーのスタッフ割り当てを使用したい場合は、次のようにします。

1. アクティビティーのプロパティー・ウィンドウをオープンします。
2. 「**制御 (Control)**」タブをクリックします。
3. 「**プロセス割り当ての組み込み (Include process assignment)**」をクリックします。

これはスタッフ定義の基準がアクティビティー・プロパティーではなく、プロセス・プロパティーから取られることを定義します。これは、アクティビティーに有効なプロセスから定義をとろうとする場合に有利です。

アクティビティーには、42ページの図9 で示されているように、「**2 ページからの動的割り当て (Dynamic assignment from page 2)**」を選択しなければなりません。

継承するスタッフ解決の定義

ワークフロー・モデルでは、ドメイン全体に有効なスタッフ解決動作を指定することができます。トポロジー階層の頂点がドメインで、その次にシステム・グループ、そしてシステムへと続きます。スタッフ定義では、プロパティーの定義の仕方によって、プロセスまたはアクティビティーが最低レベルになります。階層の最高レベルにスタッフ解決を指定し、トポロジー階層の下位レベルに継承されるようにプロパティーを定義することができます。

しかし、下位レベルのプロセスまたはアクティビティーの定義は変更することができます。プロセスまたはアクティビティーの定義を変更する場合は、「**継承 (Inherited)**」オプションの選択を解除します。

プロセスのためのオプション

以下のオプションを定義することができます。ただし、これらはサブプロセスにのみ適用されます。これらのオプションを定義するには、次のようにします。

1. 「ネットワーク (**Network**)」 ツリー・ビューをオープンします。
2. 定義するものに合わせて、ドメイン、システム・グループ、またはシステムの「プロパティ (**Properties**)」をオープンします。
3. 「プロセス (**Process**)」 タブをクリックします。
4. 設定する「自律 (**Autonomy**)」 オプションを選択します。

スタッフ (**Staff**)

スタッフ自律を指定すると、以下に示すプロパティは親プロセスから独立します。

- 組織 (Organization)
- 役割 (Role)
- プロセス割り当ての組み込み (Include process assignment)
- ローカル・ユーザー優先 (Prefer local users)
- ユーザーが不在の場合の代理人割り当て (Assign substitute if user is absent)

通知 (**Notification**)

通知自律を指定すると、以下に示すプロパティは親プロセスから独立します。

- アクティビティ期間 (Activity duration)
- 通知モード - 実行 (Run) または保留 (Hold)
- ユーザーが不在の場合の通知代理人割り当て (Assign substitute for notification if user is absent)
- 同一ユーザーへの 2 番目の通知の送信 (Send second notification to same user)

管理 (**Administration**)

管理自律を指定すると、プロセス管理者は親プロセスからはとられません。

制御 (**Control**)

制御自律を指定すると、終了、中断、および再開について、親システムの要求はそのプロセスに影響しません。

アクティビティーのためのオプション

アクティビティーのための以下のオプションを指定することができます。これらのオプションを定義するには、次のようにします。

1. 「ネットワーク (Network)」 ツリー・ビューをオープンします。
 2. 定義するものに合わせて、ドメイン、システム・グループ、またはシステムの「プロパティ (Properties)」をオープンします。
 3. 「アクティビティー (Activity)」 タブをクリックします。
 4. 設定する「スタッフ割り当て (Staff assignment)」 オプションを選択します。
- 「プロセス割り当ての組み込み (Include process assignment)」については、130ページの『継承するスタッフ解決の定義』に記載されています。
 - 「ローカル・ユーザー優先 (Prefer local users)」、「非ローカル・ユーザー優先 (Prefer not absent users)」、および「ユーザーが不在の場合の代理人割り当て (Assign substitute if user is absent)」については、123ページの『アクティビティーのためのスタッフ解決の定義』に記載されています。

スタッフ解決の評価のための規則

MQSeries Workflow では、スタッフ解決のための定義は以下のように評価されます。

1. プロセスおよびアクティビティーのスタッフ解決の定義はプロセス・モデル定義の一部です。このことは、プロセスをプロセス・テンプレートに変換した後でも、そのプロセス・テンプレートにこれらの定義が設定されることを意味しています。そしてこれらの定義は実行機能で開始するすべてのプロセス・インスタンスに有効です。
2. ドメイン、システム・グループ、およびシステムに指定するオプションは、ワークフロー・モデルを実行機能に搬入する時に、適切な FDL 定義を指定することによって設定できます。これらの定義を搬入し終えてシステムを再始動すると、その変更が即時にスタッフ解決に適用され、実行時に有効になります。

スタッフ解決のパフォーマンスの考慮

スタッフ解決の定義によっては、多人数向けの作業項目を作成することができます。たとえば、それぞれのアクティビティーごとに 30 ~ 50 人向けの作業項目が作成されれば、ワークフロー・システムのパフォーマンスにかなり影響します。

通知とは？

MQSeries Workflow では、プロセス定義担当者は次の時間を指定することができます。

- プロセスが終了すべき時間
- プロセスに定義された各アクティビティーが終了すべき時間
- 通知を受けた人がそれに対して行動すべき時間

これらのプロセス、作業項目、および通知が指定された時間内に完了しないと、指定された人に通知されます。



通知の検査がいつ行われるかは、間隔設定によって決められます。間隔設定は定義機能で定義されます。

通知された作業項目、アクティビティー、およびプロセス・インスタンスの取り扱い方法についての詳細は、*IBM MQSeries Workflow: 実行機能の開始*、および *MQSeries Workflow* クライアントのオンライン・ヘルプを参照してください。

定義機能では、実行時に通知作業項目を受信する人を定義することができます。

- 通知がアクティビティーに設定されていて、アクティビティーを完了する時間が指定された時間を超えた場合、指定された人は通知作業項目を受け取ります。さらに、2 番目の通知を指定することができます。通知が指定された時間内に完了しない場合、2 番目の通知がプロセス管理者に送られます。
- 通知がプロセス・プロパティーの「**スタッフ (Staff)**」ページにあるプロセスに設定されている場合は、プロセス管理者が通知を受け、2 番目の通知はありません。

アクティビティーのプロパティーを基にした通知

アクティビティーのための通知動作を定義するには、「**通知 (Notification)**」ページにあるオプションを選択することができます。通知は定義された基準を満たした人に割り当てられます。次のように指定できます。

定義済みメンバーからの通知 (Notification from predefined members)

通知作業項目は、コンテナー・メンバー

`_ACTIVITY_INFO.PersonToNotify` で指定された人に割り当てられます。

コンテナー・メンバーの詳細については、*IBM MQSeries Workflow: プログラミングの手引き* を参照してください。

遅延通知人 (Person to notify of delay)

「なし (None)」を選択すると、通知は作成されません。

プロセス管理者 (process administrator)

通知作業項目はプロセス管理者に直接割り当てられます。

マネージャー (Manager)

通知作業項目は、元の作業項目が割り当てられたすべての人のマネージャーに割り当てられます。

コーディネーター (Coordinator)

このオプションは、プロセスおよびアクティビティの「スタッフ 1 (Staff 1)」プロパティ・ページにある「役割のコーディネーター (Coordinator of role)」オプションを指定した場合のみ適用されます。指定されているなら、元のアクティビティのスタッフ解決に用いられたすべての役割のコーディネーターが通知作業項目を受け取ります。

人 (Person)

通知作業項目は指定された人に割り当てられます。

コンテナにある (From container)

通知作業項目はコンテナ・メンバーの中で指定された人に割り当てられます。

通知期間 (Duration of notification)

これは、アクティビティを作業可能 (Ready) 状態に設定してからアクティビティが完了するまでの時間間隔を定義します。この時間内にアクティビティが完了しないと通知が作成されることとなります。この時間間隔は明確に定義するか、コンテナ・メンバーからとることができます。時間間隔がコンテナ・メンバーからとられる場合、その値は、通知が作成されるまでの秒数として解釈されます。

プロセスが中断した場合、通知方式の設定によっては、通知タイマーが停止することがあります。

決定の期間 (Duration of making decision)

2 番目の通知は通知アクティビティを作動可能 (Ready) 状態に設定してからアクティビティが完了するまでの時間間隔です。この間にアクティビティが完了しないと、2 番目の通知が作成されます。この時間間隔は明確に定義するか、コンテナ・メンバーからとることができます。時間間隔がコンテナ・メンバーからとられる場合、その値は、2 番目の通知が作成されるまでの秒数として解釈されます。

通知オプションを定義するには、「**制御 (Control)**」ページのオプションを選択します。次のように指定できます。

ユーザーが不在の場合の通知代理人割り当て (Assign substitute for notification if user is absent)

不在として宣言されていないユーザーのみが通知作業項目を受信します。ユーザーが不在の場合、通知作業項目は代理人に割り当てられます。これは、代理人が不在として宣言されていない場合にのみ適用されます。適格ユーザーがいない場合、作業項目はプロセス管理者に割り当てられます。

同一ユーザーへの 2 番目の通知送信 (Send second notification to same user) 2 番目の通知が必要な場合は、プロセス管理者に割り当てられます。最初の通知がプロセス管理者に割り当てられた場合は、このオプションが指定されている場合にも、2 番目の通知が送信されます。

プロセスのプロパティを基にした通知

プロセスの通知動作を定義するには、プロセスのプロパティの「**スタッフ (Staff)**」ページにあるオプションを選択します。通知は定義された基準を満たした人に割り当てられます。次のように指定できます。

プロセスの期間 (Duration of process)

プロセスの通知時間を明確に指定するか、コンテナ・メンバーからとられるように指定することができます。プロセスの期間はプロセスの開始から終了または中止までの時間枠を指定します。指定された時間枠内にプロセスが終了しないと、プロセス通知作業項目がプロセス管理者に割り当てられます。

「**制御 (Control)**」ページでは、以下を指定することができます。

通知モード (Notification mode)

以下はプロセス通知およびアクティビティ通知に対して有効です。

「**実行 (Run)**」または「**保留 (Hold)**」を選択して、プロセスが中断した場合のプロセスの通知タイマーの動作を定義することができます。

1. 「**保留 (Hold)**」を選択すると、中断の間、タイマーは停止します。
2. 「**実行 (Run)**」を選択すると、中断の間もタイマーは稼働し続けます。

以下のオプションはプロセスのすべてのアクティビティに適用されます。

「**継承 (Inherited)**」が選択されると、アクティビティに対してその値が与えられます。継承についての詳細は、130ページの『継承するスタッフ解決の定義』の以下の項目を参照してください。

- ユーザーが不在の場合の通知代理人割り当て (Assign substitute for notification if user is absent)
- 同一ユーザーへの 2 番目の通知の送信 (Send second notification to same user)

ネットワーク用に定義された通知

130ページの『継承するスタッフ解決の定義』で説明されているとおり、MQSeries Workflow ドメイン、システム・グループ、およびシステム全体に対する通知動作を定義することもできます。

「プロセス (Process)」ページでは以下を指定することができます。

通知モード (Notification mode)

「実行 (Run)」または「保留 (Hold)」を指定できます。

「アクティビティ (Activity)」ページでは、以下を指定することができます。

- ユーザーが不在の場合の通知代理人割り当て (Assign substitute for notification if user is absent)
- 同一ユーザーへの 2 番目の通知の送信 (Send second notification to same user)

ワークフロー・モデルに適用される権限オプションについての詳細は、*IBM MQSeries Workflow: プログラミングの手引き* の権限の考慮事項に関する章を参照してください。プロセスをプロセス・テンプレートに変換するときに検査されるものの詳細については、51ページの『ワークフロー・モデルの検証』を参照してください。

付録B. 定義機能データベースの再編成

他のリレーショナル・データベースと同様、定義機能データベースは定期的に再編成して、サイズを減らす必要があります。これにより、ディスク・スペースを使い尽くさないようにすることができます。

リレーショナル・データベースを使用しているとき、エントリーを追加すればディスク・スペースの使用量が増加しますが、エントリーを削除してもディスク・スペースの使用量は減りません。したがって、新しいスペースを確保し、データベースのサイズを減らすには、データベースをコンパクト化する必要があります。

データベースの再編成およびコンパクト化を開始する前に、通常のバックアップ手順で使用するツールで、データベースをバックアップしてください。

定義機能と IBM DB2 ユニバーサル・データベース (DB2 Universal Database (R))

定義機能のために DB2 (R) データベースを使用している場合には、『DB2 管理の手引き』に示されている方法でデータベースを再編成できます。DB2 フォルダーの管理ツールは、DB2 インストールの一部であり、DB2 データベースを再編成するために使用できます。

Microsoft Jet データベース・エンジンの使用

データベースを再編成するには、インストールの一部である **ODBC データ・ソース管理機能 (ODBC Data Source Administrator)** を使用することができます。

定義機能データベースのファイル名は定義機能の構成中に定義されます。構成中にデフォルト設定を使用した場合、データベースは MQSeries Workflow インストール・パスの **bt_db** というディレクトリーに配置されます。ファイル名は **fmcbtdb.mdb** です。

ODBC データ・ソース管理機能 (ODBC Data Source Administrator) を開始するには、次のようにします。

1. 「**コントロール パネル**」をオープンします。
2. 「**ODBC**」をダブルクリックします。

「**ODBC**」アイコンが見つからない場合には、Windows NT または Windows 95 のシステム・ディレクトリーに **ODBCAD32.EXE** が存在するかどうかをチェックしてください。このプログラムは、定義機能インストールで自動的にインストールされます。**ODBCAD32.EXE** を開始します。これにより、「**ODBC データ・ソース管理機能 (ODBC Data Source Administrator)**」がオープンします。

3. 「**システム DSN (System DSN)**」タブまたは「**ユーザー DSN (User DSN)**」タブでデータベース名 **fmcbtdb** を探し、データベース **fmcbtdb** の名前をクリックします。

4. 「**構成 (Configure)**」をクリックします。

これにより、「**ODBC Microsoft Access 97 セットアップ**」がオープンします。このダイアログ・ボックスおよび後続のダイアログ・ボックスに正しいデータベース名が自動的に表示されます。

5. 「**コンパクト化 (Compact)**」をクリックします。

6. 「**コンパクト化する元のデータベース (Database To Compact From)**」ダイアログ・ボックスで「**OK**」をクリックします。

7. 「**コンパクト化の出力先データベース (Database To Compact Into)**」ダイアログ・ボックスで「**OK**」をクリックします。

次のような警告メッセージが表示されます。 The database already exists. (データベースがすでに存在します。) Do you want to replace it? (置き換えますか?)

8. 「**はい (Yes)**」をクリックします。

次のような通知メッセージが表示されます。 The database was successfully compacted. (データベースは正常にコンパクト化されました。)

9. 「**OK**」をクリックして、データベースの再編成を終了します。

付録C. 特記事項

本書において、日本では発表されていない IBM 製品 (機械およびプログラム)、プログラミングまたはサービスについて言及または説明する場合があります。しかし、このことは、弊社がこのような IBM 製品、プログラミングまたはサービスを、日本で発表する意図があることを必ずしも示すものではありません。本書で IBM ライセンス・プログラムまたは他の IBM 製品に言及している部分があっても、このことは当該プログラムまたは製品のみが使用可能であることを意味するものではありません。IBM 製品、プログラム、またはサービスに代えて、IBM の有効な知的所有権またはその他の法的に保護された権利を侵害することのない、機能的に同等の製品、プログラム、またはサービスを使用することができます。ただし、IBM によって明示的に指定されたものを除き、他社の製品と組み合わせた場合の操作の評価と検証はお客様の責任で行っていただきます。

IBM は、本書で解説されている主題について特許権 (特許出願を含む)、商標権、または著作権を所有している場合があります。本書の提供は、これらの特許権、商標権、および著作権について、本書で明示されている場合を除き、実施権、使用権等を許諾することを意味するものではありません。実施権、使用権等の許諾については、下記の宛先に、書面にてご照会ください。

〒106-0032 東京都港区六本木 3 丁目 2-31
AP 事業所
IBM World Trade Asia Corporation
Intellectual Property Law & Licensing

以下の保証は、国または地域の法律に沿わない場合は、適用されません。IBM およびその直接または間接の子会社は、本書を特定物として現存するままの状態を提供し、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任を負わないものとします。国または地域によっては、法律の強行規定により、保証責任の制限が禁じられる場合、強行規定の制限を受けるものとします。

本書に対して、周期的に変更が行われ、これらの変更は、文書の次版に組み込まれます。IBM は、随時、この文書に記載されている製品またはプログラムに対して、改良または変更を行うことがあります。

本プログラムのライセンス保持者で、(i) 独自に作成したプログラムとその他のプログラム (本プログラムを含む) との間での情報交換、および (ii) 交換され

た情報の相互利用を可能にすることを目的として、本プログラムに関する情報を必要とする方は、下記に連絡してください。

IBM Deutschland
Informationssysteme GmbH
Department 3982
Pascalstrasse 100
70569 Stuttgart
Germany

本プログラムに関する上記の情報は、適切な使用条件の下で使用することができますが、有償の場合もあります。

本書で説明されているライセンス・プログラムまたはその他のライセンス資料は、IBM 所定のプログラム契約の契約条項、IBM プログラムのご使用条件、またはそれと同等の条項に基づいて、IBMより提供されます。

この文書に含まれるいかなるパフォーマンス・データも、管理環境下で決定されたものです。そのため、他の操作環境で得られた結果は、異なる可能性があります。一部の測定が、開発レベルのシステムで行われた可能性がありますが、その測定値が、一般に利用可能なシステムのものと同じである保証はありません。さらに、一部の測定値が、推定値である可能性があります。実際の結果は、異なる可能性があります。お客様は、お客様の特定の環境に適したデータを確かめる必要があります。

IBM 以外の製品に関する情報は、その製品の供給者、出版物、もしくはその他の公に利用可能なソースから入手したものです。IBM は、それらの製品のテストは行っておりません。また、IBM 以外の製品に関するパフォーマンスの正確性、互換性、またはその他の要求は確認できません。IBM 以外の製品の性能に関する質問は、それらの製品の供給者にお問い合わせください。

IBM の将来の方向または意向に関する記述については、予告なしに変更または撤回される場合があります、単に目標を示しているものです。

これらの情報は、計画の目的でのみ提供するものです。この情報は、説明されている製品が一般的に提供可能になる前に変更される場合があります。

本書には、日常の業務処理で用いられるデータや報告書の例が含まれています。より具体性を与えるために、それらの例には、個人、企業、ブランド、あるいは製品などの名前が含まれている場合があります。これらの名称はすべて架空のものであり、名称や住所が類似する企業が実在しているとしても、それは偶然にすぎません。

著作権使用許諾：

本書には、様々なオペレーティング・プラットフォームでのプログラミング手法を例示するサンプル・アプリケーション・プログラムがソース言語で掲載されています。お客様は、サンプル・プログラムが書かれているオペレーティング・プラットフォームのアプリケーション・プログラミング・インターフェースに準拠したアプリケーション・プログラムの開発、使用、販売、配布を目的として、いかなる形式においても、IBM に対価を支払うことなくこれを複製し、改変し、配布することができます。これらの例は、すべての場合について完全にテストされたものではありません。IBM はこれらのプログラムの信頼性、可用性、および機能について法律上の瑕疵担保責任を含むいかなる明示または暗示の保証責任も負いません。

サンプル・ソース・コードのすべての部分、またはすべての派生した創作物には、次のように、著作権表示を入れていただく必要があります。

© (お客様の会社名) (西暦年). このコードの部分は、IBM Corp. のサンプル・プログラムからとったものです。

© Copyright IBM Corp. 1993, 2000. All rights reserved.

この情報をソフトコピーでご覧になっている場合には、写真およびカラーの図表は現れない場合があります。

商標

次のものは、IBM Corporation の商標です。

- AIX
- DB2
- DB2 Universal Database
- FlowMark
- IBM
- MQSeries
- OS/2
- OS/390
- RISC System/6000

Lotus Notes は Lotus Development Corporation の登録商標であり、Domino および Lotus Go Webserver は同社の商標です。

Microsoft、Windows、Windows NT、および Windows ロゴは Microsoft Corporation の米国およびその他の国における商標です。

UNIX は、The Open Group がライセンスしている米国およびその他の国における登録商標です。

ActionMedia、LANDesk、MMX、Pentium、および ProShare は Intel Corporation の米国およびその他の国における商標です。

Java およびすべての Java 関連の商標およびロゴは Sun Microsystems, Inc. の米国およびその他の国における商標または登録商標です。

他の会社名、製品名およびサービス名等はそれぞれ各社の商標または登録商標です。

用語集

この用語集には、本書で使用している重要語および省略語の定義を記載しています。探している用語が見つからない場合には、索引か、*IBM Dictionary of Computing* (New York: McGraw-Hill, 1994) を参照してください。

A

管理サーバー (administration server).

MQSeries Workflow のうち、MQSeries Workflow システム内で管理機能を実行する構成要素。その機能には、MQSeries Workflow システムの起動 / 停止、エラー管理の実行、システム・グループの管理機能への参加が含まれる。

アクティビティー (activity). プロセス・モデルを構成するステップの 1 つ。プログラム・アクティビティー、プロセス・アクティビティー、ブロック・アクティビティーのいずれか。

アクティビティー情報メンバー (activity information member). アクティビティーの操作特性に関する定義済みデータ構造メンバー。

API. アプリケーション・プログラミング・インターフェース (application programming interface)。

アプリケーション・プログラミング・インターフェース (application programming interface). プログラムで MQSeries Workflow のワークフロー・マネージャーのサービスを要求するのに使うインターフェースとして MQSeries Workflow のワークフロー・マネージャーに用意されているもの。サービスは、同期的に提供される。

監査証跡 (audit trail). データベースの中のリレショナル表の 1 つで、プロセス・インスタン

スの実行中に発生する各イベントごとに 1 つの項目が含まれているもの。

許可 (authorization). MQSeries Workflow におけるユーザーの権限レベルを決定する、ユーザー・スタッフ定義の属性。システム管理者はすべての機能を実行できる。

B

屈曲点 (bend point). コネクターが開始したり終了したり方向を変更したりする点。

ブロック・アクティビティー (block activity). 一群のアクティビティーから成る複合アクティビティー。制御およびデータ・コネクターに接続可能。ブロック・アクティビティーは Do-Until ループを実現するために使われる。ブロック・アクティビティー内のすべてのアクティビティーは、ブロック・アクティビティーの終了条件が満たされるまで処理される。複合アクティビティー (composite activity) も参照。

定義機能 (Buildtime). MQSeries Workflow の構成要素の 1 つで、グラフィカル・ユーザー・インターフェースが含まれており、ワークフロー・モデルの作成 / 保守、リソースの管理、およびシステム・ネットワークの定義のために使用するもの。

C

カーディナリティー (cardinality). (1) 関係の属性で、メンバー条件の数量について述べているもの。1 対 1、1 対多、多対 1、多対多の 4 種類カーディナリティーがある。(2) データベース表の中の行数、またはデータベース表の 1 列の中の互いに異なる値の数。

下位組織 (child organization). 企業の管理単位の階層に含まれる組織のうち、上位組織のある組織。1つの下位組織に対して、上位組織が1つ、そして下位組織が複数個可能。上位組織とは、階層内で1レベル上位の組織のこと。上位組織 (*parent organization*) と対比。

クリーンアップ・サーバー (cleanup server). MQSeries Workflow の構成要素で、論理的にしか削除されていない情報を MQSeries Workflow 実行機能データベースから物理的に削除するためのもの。

複合アクティビティー (composite activity). いくつかのアクティビティーから構成されるアクティビティー。複合アクティビティーは、ブロック・アクティビティーかバンドル・アクティビティーである。

コンテナ API (container API). MQSeries Workflow の制御下で実行されているプログラムがアクティビティーの入力および出力コンテナからデータを入手したり、アクティビティーの出力コンテナにデータを格納したりするための MQSeries Workflow API。

制御コネクタ (control connector). プロセスに含まれる2つのノードの間で制御の潜在的な流れを定義するもの。実際の制御の流れは、制御コネクタに関連付けられた分岐条件の真理値によって実行時に決まる。

コーディネーター (coordinator). 役割を調整する人として指定された担当者に対して自動的に割り当てられる定義済みの役割。

D

データ・コネクタ (data connector). コンテナとコンテナの間のデータの流れを定義するもの。

データ・コンテナ (data container). アクティビティーまたはプロセスの入力データおよび出

カデータのストレージ。入力コンテナ (*input container*) および出力コンテナ (*output container*) を参照。

データ・マッピング (data mapping). データ・コネクタについて、関連するソース・コンテナのフィールドと関連するターゲット・コンテナのフィールドの対応を指定するもの。

データ構造 (data structure). データ構造メンバーの集合からなる名前付きエンティティー。入力および出力コンテナはデータ構造への参照により定義され、参照先のデータ構造タイプのレイアウトを採用する。

データ構成メンバー (data structure member). データ構造を構成する変数の1つ。

デフォルト制御コネクタ (default control connector). プロセス・ダイアグラムで標準の制御コネクタを視覚的に表現するもの。他の制御パスが有効でなければ、制御はこのコネクタを流れる。

ドメイン (domain). メタモデルが同じで、同じスタッフ情報とトポロジー情報を共用する MQSeries Workflow システム・グループの集合。ドメイン内の構成要素間の通信は、メッセージ・キューイングを使ってなされる。

動的スタッフ割り当て (dynamic staff assignment). 役割、組織、またはレベルなどの基準を指定することによりスタッフをアクティビティーに割り当てる手法。アクティビティーが作動可能になったときに、選択基準に合うユーザーが、作業対象のアクティビティーを受け取る。レベル (*level*)、組織 (*organization*)、プロセス管理者 (*process administrator*)、および役割 (*role*) も参照。

E

終了アクティビティー (end activity). 出ていく制御コネクタがないアクティビティー。

実行サーバー (execution server). MQSeries Workflow の構成要素で、実行時にプロセス・インスタンスの処理を実行するもの。

終了条件 (exit condition). アクティビティーが完了したかどうかを指定する論理式。

搬出 (export). MQSeries Workflow データベースから情報を検索し、それを MQSeries Workflow 定義言語 (FDL) または HTML 形式でできるようにする MQSeries Workflow のユーティリティー・プログラム。搬入 (*import*) と対比。

F

固定メンバー (fixed member). 現行アクティビティーに関する情報を提供する定義済みデータ構造メンバー。固定メンバーの値は、MQSeries Workflow のワークフロー・マネージャーによって設定される。

MQSeries Workflow 定義言語 (FDL) ((FDL) MQSeries Workflow Definition Language).

MQSeries Workflow システム・グループ相互間で MQSeries Workflow 情報を交換するための言語。この言語は MQSeries Workflow の搬出 / 搬入機能で使用される。この言語にはスタッフ、プログラム、データ構造、およびトポロジーに関するワークフロー定義が含まれる。これにより、非 MQSeries Workflow 構成要素が MQSeries Workflow と対話することが可能になる。搬出 (*export*) および搬入 (*import*) も参照。

フォーク・アクティビティー (fork activity). 複数の制御コネクタの元になっているアクティビティー。

フォーム (form). Lotus Notes において、フォームによって、Lotus Notes に情報を入力する方法、およびその情報の表示と印刷の方法が制御される。

式 (formula). Lotus Notes において、たとえばデータベースから文書を選択したり表示用の値を計算するのに使われる数式。

完全修飾名 (fully-qualified name). 完全に修飾された名前。つまり、その名前が指す構造メンバーより上のすべての名前を階層順に含み、さらにそのメンバー自体の名前を付けたもの。

I

搬入 (import). MQSeries Workflow 定義言語 (FDL) 形式で情報を受け取り、それを MQSeries Workflow データベースに入れる MQSeries Workflow ユーティリティー・プログラム。搬出 (*export*) と対比。

入力コンテナ (input container). アクティビティーまたはプロセスへの入力に使用されるデータのためのストレージ。ソース (*source*) およびデータ・マッピング (*data mapping*) も参照。

L

レベル (level). MQSeries Workflow データベースの各担当者に割り当てられる 0~9 の数値。定義機能でスタッフを定義する担当者は、ランクや経験などの意味をこれらの数値に割り当てることができる。レベルは、アクティビティーを担当者に動的に割り当てるために使用できる基準の 1 つ。

ローカル・ユーザー (local user). スタッフ解決において、親プロセスと同じシステム・グループにホーム・サーバーがあるユーザーのこと。

ローカル・サブプロセス (local subprocess). 親プロセスと同じ MQSeries Workflow システム・グループで処理されるサブプロセス。

論理式 (logical expression). 演算子とオペランドで構成され、評価結果が真、偽、または整数に

なる式。(0 でない整数は偽に相当する。) 終了条件 (*exit condition*) および分岐条件 (*transition condition*) も参照。

M

マネージャー (manager). 組織の長として定義された担当者に対して自動的に割り当てられる定義済みの役割。

メッセージ・キューイング (message queuing). ソフトウェア構成要素相互間の通信で非同期メッセージを使用する通信方法。

N

ナビゲーション (navigation). 1 つのプロセス内で、完了したアクティビティから、それ以降のアクティビティへ移ること。その経路は、制御コネクタとそれに関連する分岐条件によって、またアクティビティの開始条件によって決まる。制御コネクタ (*control connector*)、終了条件 (*exit condition*)、分岐条件 (*transition condition*)、および開始条件 (*start condition*) も参照。

ノード (node). (1) プロセス・ダイアグラム内のアクティビティの総称。(2) MQSeries Workflow システムに対してホストとなるオペレーティング・システムのイメージ。

通知 (notification). MQSeries Workflow の機能の 1 つで、プロセスまたはアクティビティが指定した期間内に完了しない場合に、指定された担当者に通知するもの。

通知作業項目 (notification work item). アクティビティまたはプロセスの通知を表す作業項目。

O

組織 (organization). 企業の管理単位。組織は、アクティビティを担当者に動的に割り当てるた

めに使用できる基準の 1 つ。下位組織 (*child organization*) および上位組織 (*parent organization*) を参照。

出力コンテナ (output container). 他のアクティビティが使用するために、または条件を評価するために、アクティビティまたはプロセスによって生成されるデータ用のストレージ。シンク (*sink*) も参照。

P

上位組織 (parent organization). 企業の管理単位の階層に含まれる組織のうち、1 つまたは複数の下位組織のある組織。下位組織は、階層内でのその上位組織よりも 1 レベル下位の組織。下位組織 (*child organization*) と対比。

親プロセス (parent process). あるプロセスをサブプロセスとして開始したプロセス・アクティビティが含まれるプロセス・インスタンス。

パターン・アクティビティ (pattern activity). バンドル・アクティビティ内のアクティビティのうち、実行時に複数のインスタンス (パターン・アクティビティ・インスタンス) が作成されるその作成元となる単一の単純なアクティビティ。

担当者 (person (複数形 : people)). MQSeries Workflow データベースで定義されている企業内のスタッフのメンバー。

事前定義データ構造メンバー (predefined data structure member). MQSeries Workflow で事前に定義されていて、ユーザー・アプリケーションと MQSeries Workflow 実行機能の間の通信で使用されるデータ構造メンバー。

プロセス (process). プロセス・モデル (*process model*) およびプロセス・インスタンス (*process instance*) の同義語。実際の意味は文脈によって異なる。

プロセス・アクティビティ (process activity). プロセス・モデルの一部であるアクティビティ。プロセス・アクティビティを実行すると、プロセス・モデルのインスタンスが作成または実行される。

プロセス管理者 (process administrator). 特定のプロセス・インスタンスの管理者。管理者にはプロセス・インスタンスに対してすべての操作を実行する権限がある。管理者は、スタッフ解決および通知のターゲットにもなる。

プロセス・カテゴリー (process category). プロセス・インスタンスに対していろいろな機能を実行できるユーザーの集合を制限するために、プロセス・モデル化担当者がそのプロセス・モデルに対して指定できる属性。

プロセス定義 (process definition). プロセス・モデル (*process model*) の同義語。

プロセス・ダイアグラム (process diagram). プロセス・モデルのプロパティを示すプロセスのグラフィック表示。

プロセス・インスタンス (process instance). MQSeries Workflow 実行機能で実行されるプロセスのインスタンス。

プロセス・インスタンス・リスト (process instance list). ユーザー定義の基準に従って選択またはソートされるプロセス・インスタンスの集合。

プロセス・インスタンス・モニター (process instance monitor). MQSeries Workflow クライアント構成要素で、特定のプロセス・インスタンスの状態をグラフィック表示したもの。

プロセス管理 (process management). プロセス・インスタンスに関連した MQSeries Workflow 実行機能タスク。プロセス・インスタンスの作成、開始、一時中断、再開、中止、再始動、および削除で構成される。

プロセス・モデル (process model). プロセス・モデルの中に表示されるプロセスの集合。それらのプロセスは、プロセス・ダイアグラムの中にグラフィック形式で表される。プロセス・モデルには、プロセス・アクティビティと関連付けられたスタッフ、プログラム、およびデータ構造の定義が含まれる。プロセス・モデルをプロセス・テンプレートに搬入および変換したなら、そのプロセス・テンプレートは繰り返し実行できる。ワークフロー・モデル (*workflow model*) およびプロセス定義 (*process definition*) と同義。

プロセス・モニター API (process monitor API). アプリケーションでプロセス・インスタンス・モニターの機能を実現するためのアプリケーション・プログラミング・インターフェース。

プロセス関連データ (process-relevant data). プロセス・インスタンス内のアクティビティの順序を制御するのに使われるデータ。

プロセス状況 (process status). プロセス・インスタンスの状況。

プロセス・テンプレート (process template). プロセス・インスタンスを作成する基礎として使うことのできるプロセス・モデルの固定フォーム。MQSeries Workflow 実行機能において変換済みのフォーム。プロセス・インスタンス (*process instance*) も参照。

プロセス・テンプレート・リスト (process template list). ユーザー定義の基準に従って選択またはソートされているプロセス・テンプレートの集合。

プログラム (program). プログラム・アクティビティの実装またはサポート・ツールとなる、コンピューター・ベースのアプリケーション。プログラム・アクティビティは、MQSeries Workflow プログラム登録の中でプログラムに関連付けられている論理名を使用して、実行可能プログラムを参照する。プログラム登録 (*program registration*) も参照。

プログラム・アクティビティー (program activity). 登録済みプログラムにより実行されるアクティビティー。このアクティビティーを開始すると、プログラムが呼び出される。プロセス・アクティビティー (*process activity*) と対比。

プログラム実行エージェント (program execution agent). MQSeries Workflow の構成要素で、プログラム・アクティビティー (.EXE ファイルや .DLL ファイルなど) の実装を管理するもの。

プログラム登録 (program registration). プログラムを MQSeries Workflow に登録して、それが MQSeries Workflow によって実行される際にプログラムの管理のために必要な情報がそろっているようにすること。

R

役割 (role). スタッフ・メンバーに対して定義される責任。役割は、アクティビティーを担当者に動的に割り当てるために使用できる基準の 1 つ。

S

スケジューリング・サーバー (scheduling server). 中断している作業項目の再開や期限切れプロセスの検出などのアクションのスケジュールを時刻イベントに基づいて立てるための MQSeries Workflow 構成要素。

サーバー (server). MQSeries Workflow システムを構成するサーバー (実行サーバー、管理サーバー、スケジューリング・サーバー、およびクリンアップ・サーバー)。

シンク (sink). プロセスまたはブロック・アクティビティーの出力コンテナを表すシンボル。

ソース (source). プロセスまたはブロック・アクティビティーの入力コンテナを表すシンボル。

特定リソース割り当て (specific resource assignment). ユーザー ID を指定することによって、リソースをプロセスまたはアクティビティーに割り当てる手法。

標準クライアント (standard client). プロセス・インスタンスの作成と制御、ワーク・リストおよび作業項目の処理、およびログオン・ユーザーの個人データの操作を実行する MQSeries Workflow 構成要素。

開始アクティビティー (start activity). 入ってくる制御コネクタがないアクティビティー。

開始条件 (start condition). 入ってくるすべての制御コネクタが評価された後、それらの制御コネクタが接続されているアクティビティーを開始するかどうかを判別する条件。

サブプロセス (subprocess). プロセス・アクティビティーによって開始されるプロセス・インスタンス。

代行者 (substitute). アクティビティーの割り当て先の担当者が不在であると宣言された場合に、自動的にそのアクティビティーの転送先となる担当者。

サポート・ツール (support tool). エンド・ユーザーがアクティビティーを完了するために MQSeries Workflow クライアントのワーク・リストから開始するプログラム。

シンボルによる参照 (symbolic reference). アクティビティーの記述テキストまたはプログラム登録のコマンド行パラメーターに含まれている特定のデータ項目、プロセス名、またはアクティビティー名への参照。シンボルによる参照は、データ項目の完全修飾名または `_PROCESS` か `_ACTIVITY` のいずれかを、パーセント記号 (%) で囲んだもので表される。

システム (system). MQSeries Workflow ドメインの中での MQSeries Workflow の最小単位。これは、一群の MQSeries Workflow サーバーで構成される。

システム・グループ (system group). 同じデータベースを共用する MQSeries Workflow システムの集合。

システム管理者 (system administrator). (1) すべての許可を付与された定義済みの役割。1 つの MQSeries Workflow システムの中で 1 人の担当者に割り当てることができる。(2) コンピューターのインストール時にコンピューター・システムの設計、制御、および使用の管理を担当する担当者。

T

最上位プロセス (top-level process). サブプロセスではないプロセス・インスタンスのうち、ユーザーのプロセス・インスタンス・リストまたはアプリケーション・プログラムから開始するプロセス・インスタンス。

分岐条件 (transition condition). 条件付き制御コネクタに関連付けられている論理式。これが指定されている場合、制御がそれに関連する制御コネクタを通るためには、これが真でなければならない。*制御コネクタ (control connector)* も参照。

変換 (translate). プロセス・モデルを実行機能のプロセス・テンプレートに変換するアクション。

U

ユーザー ID (user ID). MQSeries Workflow のユーザーを固有に識別する英数字ストリング。

V

検証 (verify). プロセス・モデルが完全かどうかを調べるアクション。

W

ワークフロー (workflow). ある企業のビジネス・プロセスに従って実行される一連のアクティビティ。

Workflow Management Coalition (WfMC). ワークフロー管理システムのベンダーとユーザーからなる非営利組織。この組織の目的は、ワークフロー管理システムのためのワークフロー規格を奨励し、さまざまな実装の間の相互操作性を実現すること。

ワークフロー・モデル (workflow model). プロセス・モデル (*process model*) の同義語。

作業項目 (work item). プロセス・インスタンス内のアクティビティのコンテキストで処理される作業の表示。

ユーザーの作業項目セット (work item set of a user). 1 ユーザーに割り当てられたすべての作業項目。

ワーク・リスト (worklist). ユーザーに割り当てられて、ワークフロー管理システムから取り出される作業項目のリスト。

ワーク・リスト・ビュー (worklist view). ワーク・リストの属性であるフィルター基準に従って、ユーザーの作業項目セットから選択される作業項目および通知のリスト。そのワーク・リストにソート基準が指定されていれば、ソートすることもできる。

参照文献

ここでリストされている資料を注文する際には、IBM 担当員または IBM 事業所にご連絡ください。

MQSeries Workflow 資料

ここでリストされている資料は、MQSeries Workflowのライブラリーに含まれています。

- *IBM MQSeries Workflow: List of Workstation Server Processor Groups*, (GH12-6357) は、MQSeries Workflow のプロセッサ・グループのリストです。
- *IBM MQSeries Workflow: 概説およびアーキテクチャー*, (GH88-7348) では、MQSeries Workflow の基本概念について説明しています。また MQSeries Workflow のアーキテクチャーやそれぞれの構成要素の関係について説明しています。
- *IBM MQSeries Workflow: 定義機能の開始*, (SH88-7354) では、MQSeries Workflow の定義機能の使用方法について説明しています。
- *IBM MQSeries Workflow: 実行機能の開始*, (SH88-7349) では、クライアントの実行方法について説明しています。
- *IBM MQSeries Workflow: プログラミングの手引き*, (SH88-7352) では、アプリケーション・プログラミング・インターフェース (API) について説明しています。
- *IBM MQSeries Workflow: インストールの手引き*, (SH88-7350) では、MQSeries

Workflow のインストールとカスタマイズについての情報および手順について説明しています。

- *IBM MQSeries Workflow: 管理の手引き*, (SH88-7351) では、MQSeries Workflow システムの管理方法について説明しています。

関連資料

- *Frank Leymann, Dieter Roller, Production Workflow: Concepts and Techniques* (New Jersey: Prentice Hall PTR, 1999)
- *Frank Leymann, Dieter Roller, "Workflow-based Applications", IBM Systems Journal 36, no. 1 (1997): 102-123*, インターネットでも参照できます。
<http://www.almaden.ibm.com/journal/sj361/leymann.html>
- *Workflow Handbook 1997*, (WfMC との共同出版、Peter Lawrence 編集)

索引

日本語, 数字, 英字, 特殊文字の順に配列されています。なお, 濁音と半濁音は清音と同等に扱われています。

[ア行]

アイコン

ツール・パレットのカスタマイズ 16

独自の定義 16

アクティビティ

アクティビティの順序の制御 43

シンク・コンテナ 26

スタッフの割り当て 39

ソース・コンテナ 26

ノード 26

描画用のアイコン 27

プログラム 27

プロセス 28

プロセス・ダイアグラムへの追加 26

ブロック 28

プロパティの指定 39

移動

屈曲点 33

テキスト・フィールド 33

ノード 33

オブジェクトの状況

更新 9, 50

削除マーク付き 9

新規 9

シンボル 50

不確定 9, 50

[カ行]

開始

実行機能搬出 / 搬入 63

開始 (続き)

定義機能 11

定義機能搬出 60

定義機能搬入 61

カテゴリ

定義 25

関連

表示 23

検証

規則 51

ワークフロー・モデル 51

コーディネーター

役割 22

格子へのスナップ

格子によるオブジェクトの位置決め 29

構文

終了条件および分岐条件の表記 84

条件 81

条件の評価 87

名前と文字列の構文規則 76

FDL の規則 75

構文図の読み方 73

コネクタ

アクティビティの結合 30, 31

屈曲点の追加 31

制御 30

データ 30, 44

データ・デフォルト 30

データ・ループ 30

デフォルト 30

プロセス・ダイアグラムへの追加 30

分岐条件 43

[サ行]

再編成、定義機能データベースの

DB2 ユニバーサル・データベース 137

再編成、定義機能データベースの (続き)

Microsoft Jet データベース・エンジン 137

作業域 13

削除

屈曲点 34

コネクタ 34

ノード 34

作成

アクティビティとその順序 29

コネクタ 30

スタッフ定義 19

制御コネクタ 43

組織定義 22

データ構造 36

ネットワーク定義 23

人の定義 21

プログラム登録 36

プロセス・ダイアグラム 24

プロセス・プロパティ 25

役割定義 22

レベル 21

参考文献 151

システム管理者

役割 22

ワークフローの管理 5

実行機能

オブジェクトの状況 49

実行機能データ 8

実行機能での定義機能データ 8

データベース 49

スタッフ

アクティビティへの割り当て 39

関連の表示 23

組織の定義 22

担当者の定義 20

プロセスへの割り当て 39

役割の定義 22

レベルの指定 21

スタッフ解決動作
 スタッフ解決 123

スタッフの割り当て
 スタッフの定義 20
 動的 39, 40
 特定 40

組織
 定義 22
 メンバー 22

[タ行]

担当者
 定義 21

チェック、ワークフロー・モデルの
 アクティビティについての規則 53
 制御コネクタについての規則 58
 データ構造についての規則 60
 データ・コネクタについての規則 59
 プログラム・アクティビティと
 プロセス・アクティビティに
 ついての規則 54
 プロセスとアクティビティにつ
 いての規則 52
 プロセスについての規則 51
 プロセス・アクティビティにつ
 いての規則 56
 ブロック・アクティビティにつ
 いての規則 57

データ構造
 コンテナ用 32
 定義 36
 定義のステップ 36
 デフォルト 32
 デフォルト・データ構造 35
 ネストされた 35
 メンバー 35
 ユーザー定義のメンバー 36

データの受け渡し、アクティビティ
 一間の
 データ・フロー 44

データベース
 再編成、定義機能データベースの
 137
 実行機能データベース 8
 定義機能データベース 8
 同期を保つための指針 8

データ・コネクタ
 データの流れの指定 30

データ・コンテナ
 事前定義データ構造メンバー 47
 シンク 31
 ソース 31
 定義 31
 デフォルト値の指定 47
 プロセス・ダイアグラムへの追加
 32

データ・マッピング
 起点 45
 コンテナ・マッピング 46
 事前定義データ構造メンバー 47
 ターゲット 45
 データ・コンテナ間 44
 ドラッグ・アンド・ドロップによ
 る 46

定義
 コネクタ 30
 スタッフ 19
 組織 22
 担当者 21
 データ構造 35
 ネットワーク・プロパティ 23
 プログラムの定義 36
 プロセス 19
 プロセスについてのアクティビテ
 ィー 26
 プロセス・プロパティ 25
 役割 22
 レベル 21

定義機能
 インプリメンテーションのツリ
 ー・ビュー 17
 オブジェクトの状況 50
 開始 11
 作業域 13
 状況バー 13
 詳細ビュー 14

定義機能 (続き)
 スタッフのツリー・ビュー 17
 ダイアグラム・ビュー 14
 ツールバー 14
 ツール・パレット 15
 カスタマイズ 15
 ツリー・ビュー 12
 データベース 8
 ネットワークのツリー・ビュー
 17
 搬出 7
 搬入 7
 プロセスのツリー・ビュー 17
 プロパティ・ビュー 14
 メニュー・バー 14
 ログオン 11
 FDL 形式 7

特記事項 139

ドメイン
 継承定義 23
 ドメインについての定義 23

[ナ行]

ネストされたデータ構造
 への参照 35

ノード
 コネクタを使用しての結合 31
 ダイアグラムからの削除 34
 ダイアグラム内での移動 33
 ダイアグラムへの追加 26

[ハ行]

搬出
 オブジェクトの選択 60
 オプション 66
 コマンド構文 64
 実行機能からの 62
 実行機能のエラー・コード 62
 実行機能搬出の使用 62
 定義機能からの 49, 60
 定義機能搬出の使用 60
 例 69
 FDL への定義機能搬出 60

搬入

- オプション 66
- コマンド構文 64
- 実行機能のエラー・コード 62
- 実行機能搬入の使用 62
- 実行機能への 49, 62
- 定義機能搬入の使用 61
- 定義機能へ 61
- 例 68
- FDL の定義機能搬入 61
- MS Jet データベースを使用して
いる場合の命名 61
- ビジネス・プロセス
 - アクティビティ 3
 - アプリケーション・プログラム
3
 - サブプロセス 3
 - 制御フロー 3
 - データ・フロー 3
 - ワークフローの概念 3
- 描画ツール・パレット
 - ツール・パレットの使用 26
- プログラム登録
 - オペレーティング・システム 37
 - プログラムの定義 36
- プログラム・アクティビティ
 - プロセス・ダイアグラムへの追加
27
 - プロパティ 39
- プロセス
 - カテゴリー 25
 - カテゴリーの割り当て 25
 - スタッフの定義 19
 - ダイアグラム 24
 - アクティビティの追加 26
 - オブジェクトの移動 33
 - 格子に合わせたオブジェクト
の位置決め 29
 - コネクタの追加 30
 - 作成 19
 - データ・コンテナの追加
32
 - ドラッグ・アンド・ドロップ
によるオブジェクトの追加
28
 - ノード 26

プロセス (続き)

- ダイアグラム 24 (続き)
 - 描画 24
 - 描画に関する指針 29
 - 部分削除 34
 - 部分の切り取りと貼り付け
34
 - 部分のコピーと貼り付け 33
 - プロセス名 25
 - プロパティ 32
 - 保管 28
 - フローの定義 24
 - プロパティの指定 25
- プロセス定義
 - ダイアログ 3
 - プロセス・ダイアグラム 3
 - プロパティ 3
 - ワークフローの構成要素 4
- プロセス・アクティビティ
 - プロセス・ダイアグラムへの追加
28
- プロセス・インスタンス
 - プロセスのインスタンス 7
- プロセス・セグメントの切り取りと
貼り付け 34
- プロセス・セグメントのコピーと貼
り付け 33
- プロセス・テンプレート
 - 作成 49
 - 実行機能 での使用。 7
- ブロック・アクティビティ
 - プロセス・ダイアグラムへの追加
28
- 変換
 - および検証 69
 - 使用するユーティリティ 70
 - 搬入 69
 - ワークフロー・モデル 6

[マ行]

- マネージャー
 - 役割 22
- モデル定義のステップ
 - 推奨ステップ 5

[ヤ行]

役割

- 定義 22
- ユーザー定義
 - アイコン 16
- ユーティリティ
 - 実行機能搬出 62
 - 実行機能搬出および搬入の使用
62
 - 実行機能搬入 62
 - 搬出 49
 - 搬入 49

[ワ行]

ワークフロー・モデル

- 構成要素 3
- 実行機能用のプロセス・テンプレ
ート 6
- 紹介 3
- 定義機能と実行機能の間での転送
49
- 変換と検証 6
- モデル定義のステップ 5
- ワークリスト
 - アクティビティ 8

F

FDL (MQSeries Workflow 定義言語)

- 外部形式 73
- 形式 89
- 構文図の読み方 73
- 実行機能の搬出先 62
- 実行機能の搬入元 62
- 定義機能の搬出先 60
- 定義機能の搬入元 61
- バージョン 3.1 または 3.1.1 の使
用 70
- FDL ソース・ファイルの形式
89
- FDL の構文規則 75
- FlowMark バージョン 2.3 の使用
70

FDL 定義

- 共通変数 115
- スタッフ 100
- ツール・セット 114
- データ構造 102
- トポロジー 92
- プログラム 103
- プロセス 106
- プロセス・モデル定義 102

H

HTML 形式

- 搬出 60

M

MQSeries Workflow システム

- アーキテクチャー 6



部品番号: CT8L0JA
プログラム番号: 5697-FM3

Printed in Japan

SH88-7354-04



日本アイ・ビー・エム株式会社
〒106-8711 東京都港区六本木3-2-12

(1P) P/N: CT8L0JA



Spine information:



IBM MQSeries Workflow

定義機能の開始

バージョン 3.2.2