



# TERASOLUNA Batch Framework for Java

## 設定ファイル説明書

第 2.0.1.1 版

株式会社 NTT データ

本ドキュメントを使用するにあたり、以下の規約に同意していただく必要があります。同意いただけない場合は、本ドキュメント及びその複製物の全てを直ちに消去又は破棄してください。

1. 本ドキュメントの著作権及びその他一切の権利は、NTT データあるいは NTT データに権利を許諾する第三者に帰属します。
2. 本ドキュメントの一部または全部を、自らが使用する目的において、複製、翻訳、翻案することができます。ただし本ページの規約全文、および NTT データの著作権表示を削除することはできません。
3. 本ドキュメントの一部または全部を、自らが使用する目的において改変したり、本ドキュメントを用いた二次的著作物を作成することができます。ただし、「TERASOLUNA Batch Framework for Java（設定ファイル説明書）」あるいは同等の表現を、作成したドキュメント及びその複製物に記載するものとします。
4. 前2項によって作成したドキュメント及びその複製物を、無償の場合に限り、第三者へ提供することができます。
5. NTT データの書面による承諾を得ることなく、本規約に定められる条件を超えて、本ドキュメント及びその複製物を使用したり、本規約上の権利の全部又は一部を第三者に譲渡したりすることはできません。
6. NTT データは、本ドキュメントの内容の正確性、使用目的への適合性の保証、使用結果についての確信や信頼性の保証、及び瑕疵担保義務も含め、直接、間接に被ったいかなる損害に対しても一切の責任を負いません。
7. NTT データは、本ドキュメントが第三者の著作権、その他如何なる権利も侵害しないことを保証しません。また、著作権、その他の権利侵害を直接又は間接の原因としてなされる如何なる請求（第三者との間の紛争を理由になされる請求を含む。）に関しても、NTT データは一切の責任を負いません。

本ドキュメントで使用されている各社の会社名及びサービス名、商品名に関する登録商標および商標は、以下の通りです。

Java, JDK, J2SE, J2EE は、米国 Sun Microsystems, Inc.の米国及びその他の国における登録商標または商標です。

Oracle は、米国 Oracle International Corp.の米国及びその他の国における登録商標または商標です。

TERASOLUNA は、株式会社 NTT データの登録商標です。

WebLogic は、BEA Systems Inc.の登録商標または商標です。

その他の会社名、製品名は、各社の登録商標または商標です。

本書は、TERASOLUNA Batch Framework for Java ver2.0.1.1 に対応しています。

## Bean定義ファイル一覧

項番	名称	ファイル名
1	<a href="#">フレームワークBean定義ファイル</a>	FrameworkBean.xml
2	<a href="#">ジョブBean定義ファイル</a>	任意
3	<a href="#">デフォルトBean定義ファイル</a>	DefaultValueBean.xml
4	<a href="#">データアクセスBean定義</a>	dataAccessContext-batch.xml
5	<a href="#">非同期型ジョブ起動機能バッチデーモン</a>	AsyncBatchDaemonBean.xml
6	<a href="#">デーモン終了用ジョブBean定義ファイル</a>	StopDaemonBean.xml
7	<a href="#">非同期型起動用スレッドプール定義ファイル</a>	ThreadPoolContext-AsyncBatch.xml
8	<a href="#">同期型起動用スレッドプール定義ファイル</a>	ThreadPoolContext-batch.xml
9	<a href="#">非同期型ジョブ起動機能バッチデーモン(CommonJ用)定義ファイル</a>	AsyncBatchDaemonForCommonjBean.xml
10	<a href="#">ジョブスターター用ワークマネージャ定義ファイル</a>	WorkManagerContext.xml
11	<a href="#">ワーク用ワークマネージャ定義ファイル</a>	WorkManagerTaskContext-batch.xml
12	<a href="#">プレースホルダ設定定義ファイル</a>	PlaceholderConfig.xml
13	<a href="#">チャンク別トランザクションモデル</a>	ChunkTransactionBean.xml
14	<a href="#">チャンク別トランザクションモデル (リスタート)</a>	ChunkTransactionForRestartBean.xml
15	<a href="#">チャンク別トランザクションモデル (ブレイク)</a>	ChunkTransactionForControlBreakBean.xml
16	<a href="#">非トランザクションモデル</a>	NoTransactionBean.xml
17	<a href="#">単一トランザクションモデル</a>	SingleTransactionBean.xml
18	<a href="#">ジョブ分割チャンク別トランザクションモデル</a>	PartitionChunkTransactionBean.xml
19	<a href="#">ジョブ分割チャンク別トランザクションモデル (リスタート)</a>	PartitionChunkTransactionForRestartBean.xml
20	<a href="#">ジョブ分割非トランザクションモデル</a>	PartitionNoTransactionBean.xml
21	<a href="#">ジョブ分割単一トランザクションモデル</a>	PartitionSingleTransactionBean.xml
22	<a href="#">ジョブ分割逐次単一トランザクションモデル</a>	SequentialSingleTransactionBean.xml
23	<a href="#">ファイルアクセス</a>	FileAccessBean.xml

フレームワークBean定義ファイル

項番	設定項目名称	Bean ID	Classes/parent名	Bean概要	property	Property概要
1	ブレースホルダ				PlaceholderConfig.xml	ブレースホルダを定義したBean定義ファイル
2	デフォルトBean定義ファイルのインポート				DefaultValueBean.xml	デフォルト値を定義したBean定義ファイル
3	リソース定義	messageSource	org.springframework.context.support.ResourceBundleMessageSource	リソースバンドルのBean定義。	basenames	リソースの設定。 初期設定値は『ApplicationResources, system-messages, application-messages』である。
4		messageAccessor	jp.terasoluna.fw.batch.springsupport.messages.MessageAccessorImpl	メッセージ取得用クラスの設定	-	-
5	分割キー取得用コレクタ定義	IBatisDbPartitionKeyCollector	jp.terasoluna.fw.batch.ibatissupport.IBatisDbCollectorImpl	データベースCollectorのPartitionKey取得用Bean定義。 ジョブBean定義ファイルでCollectorに設定されたDAOを使って、データベースからPartitionKeyデータを取得する。	queryDAO	使用するQueryDao。 初期設定値は『QueryDAO』である。
6		listPropertyPartitionKeyCollector	jp.terasoluna.fw.batch.standard.ListPropertyCollector	Bean定義リストCollectorのPartitionKey取得用Bean定義。 ジョブBean定義ファイルに設定されたリストを使ってPartitionKeyデータを取得する。	collectedDataHandlerFactory	分割キー処理用ハンドラのファクトリクラス。 初期設定値は『jp.terasoluna.fw.batch.partition.PartitionKeyHandlerFactory』である。
7		stringArrayPropertyPartitionKeyCollector	stringArrayPropertyCollector	文字列配列プロパティCollectorのPartitionKey取得用Bean定義。 ジョブBean定義ファイルに設定された文字列配列を使って、PartitionKeyデータを取得する。	collectedDataHandlerFactory	分割キー処理用ハンドラのファクトリクラス。 初期設定値は『jp.terasoluna.fw.batch.partition.PartitionKeyHandlerFactory』である。
8	対象データ取得用コレクタ定義	listPropertyCollector	jp.terasoluna.fw.batch.standard.ListPropertyCollector	Bean定義リストCollector用Bean定義。 ジョブBean定義ファイルに設定されたリストを使って対象データを取得する。	collectedDataHandlerFactory	チャンカーを生成するファクトリクラス。 初期設定値は『jp.terasoluna.fw.batch.standard.ChunkerFactory』である。
9		stringArrayPropertyCollector	stringArrayPropertyCollector	文字列配列プロパティCollector用Bean定義。 ジョブBean定義ファイルに設定された文字列配列を使って対象データを取得する。	collectedDataHandlerFactory	チャンカーを生成するファクトリクラス。 初期設定値は『jp.terasoluna.fw.batch.standard.ChunkerFactory』である。
10		IBatisDbChunkCollector	jp.terasoluna.fw.batch.ibatissupport.IBatisDbCollectorImpl	データベースCollector用Bean定義。 ジョブBean定義ファイルでCollectorに設定されたDAOを使って、データベースから対象データを取得する。	queryDAO	使用するQueryDao。 初期設定値は『QueryDAO』である。
					collectedDataHandlerFactory	チャンカーを生成するファクトリクラス。 初期設定値は『jp.terasoluna.fw.batch.standard.ChunkerFactory』である。
			jp.terasoluna.fw.batch.standard.ChunkerFactory	チャンカー作成クラスの定義。	chunkSize	デフォルトBean定義ファイルを参照 ジョブBean定義ファイルを参照
11	入力チェック対象データ取得用コレクタ定義	validationListPropertyCollector	jp.terasoluna.fw.batch.standard.ListPropertyCollector	Bean定義リストCollector用Bean定義。 ジョブBean定義ファイルに設定されたリストを使って対象データを取得する。	collectedDataHandlerFactory	チャンカーを生成するファクトリクラス。 初期設定値は『jp.terasoluna.fw.batch.validation.ValidationExecutorFactory』である。
12		validationStringArrayPropertyCollector	stringArrayPropertyCollector	文字列配列プロパティCollector用Bean定義。 ジョブBean定義ファイルに設定された文字列配列を使って対象データを取得する。	collectedDataHandlerFactory	チャンカーを生成するファクトリクラス。 初期設定値は『jp.terasoluna.fw.batch.validation.ValidationExecutorFactory』である。
13		validationIBatisDbChunkCollector	jp.terasoluna.fw.batch.ibatissupport.IBatisDbCollectorImpl	データベースCollector用Bean定義。 ジョブBean定義ファイルでCollectorに設定されたDAOを使って、データベースから対象データを取得する。	queryDAO	使用するQueryDao。 初期設定値は『QueryDAO』である。
					collectedDataHandlerFactory	チャンカーを生成するファクトリクラス。 初期設定値は『jp.terasoluna.fw.batch.validation.ValidationExecutorFactory』である。
			jp.terasoluna.fw.batch.validation.ValidationExecutorFactory	入力チェック用の対象データ取得ハンドラ (CollectedDataHandler) クラスの定義。 チャンク作成処理は、 collectedDataHandlerFactoryに設定されたチャンカー生成	collectedDataHandlerFactory	チャンカーを生成するファクトリクラス。 初期設定値は『jp.terasoluna.fw.batch.standard.ChunkerFactory』である。
14					validator	バリデータの定義。
15					validationResultHandler	入力チェック用XML設定ファイルのバスの定義。
16	マネージャ定義	baseManager	jp.terasoluna.fw.batch.core.JobManager	ジョブ等のフレームワークで規定する作業単位を管理、実行するマネージャ用Bean定義。	collectorResultHandler, exceptionHandlerMap	対象データ取得結果のハンドラBeanの定義。 初期設定値は『CollectorResultHandler』である。 例外ハンドラのBean定義。 初期設定値は『ExceptionHandlerMap』である。
17		baseWorker	jp.terasoluna.fw.batch.core.JobWorker	Chunkを処理単位として、Chunkが持つデータを入力とするビジネスロジックを実行するマネージャ用Bean定義。	batchUpdateResultHandler, defaultExceptionHandlerMap, throwableHandler	デフォルト例外ハンドラ用Bean定義。 初期設定値は『defaultExceptionHandler』である。 ThrowableハンドラのBean定義。 初期設定値は『throwableHandler』である。 デフォルトBean定義ファイルを参照。
18		baseQueueProcessor	jp.terasoluna.fw.batch.standard.QueueProcessor	キューから要素を取り出し、取り出したそれぞれ要素をパラメータとしてワーカを起動するマネージャ用Bean定義。	なし	
19	サポートプロセッサ	transactionalSupportProcessor	jp.terasoluna.fw.batch.springsupport.transaction.TransactionalsupportProcessor	トランザクション処理用Bean定義。 SupportProcessorインタフェースの実装クラス。	supportLogicResultHandler, defaultExceptionHandlerMap, throwableHandler	サポートロジックの処理結果ハンドラ用設定。 初期設定値は『transactionalsupportLogicResultHandler』である。 デフォルト例外ハンドラの設定。 初期設定値は『defaultExceptionHandler』である。 例外ハンドラマップの設定。 初期設定値は『ExceptionHandlerMap』である。 ThrowableハンドラのBean定義。 初期設定値は『throwableHandler』である。 デフォルトBean定義ファイルを参照。
20		standardSupportProcessor	jp.terasoluna.fw.batch.standard.StandardSupportProcessor	サポートロジックの起動用クラスのBean定義。	supportLogicResultHandler, defaultExceptionHandlerMap, throwableHandler	サポートロジックの処理結果ハンドラ用設定。 初期設定値は『supportLogicResultHandler』である。 デフォルト例外ハンドラの設定。 初期設定値は『defaultExceptionHandler』である。 例外ハンドラマップの設定。 初期設定値は『ExceptionHandlerMap』である。 ThrowableハンドラのBean定義。 初期設定値は『throwableHandler』である。 デフォルトBean定義ファイルを参照。

ジョブBean定義ファイル

項番	設定項目名称	Bean ID	Classes/parent名	概要	property	property概要	必須
1	処理モデルインポート				ChunkTransactionBean.xml	チャンク別トランザクションモデルBean定義ファイル	○ 処理モデルを一つ選び、インポートする必要がある
2					ChunkTransactionForRestartBean.xml	リスタート機能を使用するチャンク別トランザクションモデルBean定義ファイル	
3					ChunkTransactionForControlBreakBean.xml	コントロールブレイク機能を使用するチャンク別トランザクションモデルBean定義ファイル	
4					NoTransactionBean.xml	トランザクション制御なしモデルのBean定義ファイル	
5					PartitionChunkTransactionBean.xml	分割ジョブのチャンク別トランザクションモデルBean定義ファイル	
6					PartitionChunkTransactionForRestartBean.xml	リスタート機能を使用する分割ジョブのチャンク別トランザクションモデルBean定義ファイル	
7					PartitionNoTransactionBean.xml	分割ジョブのトランザクション制御なしモデルのBean定義ファイル	
8					PartitionSingleTransactionBean.xml	分割ジョブの単トランザクションモデルのBean定義ファイル	
9					SingleTransactionBean.xml	単トランザクションモデルのBean定義ファイル	
10					SequentialSingleTransactionBean.xml	ジョブ分割逐次単トランザクションモデルのBean定義ファイル	
11					FileAccessBean.xml	ファイルアクセス用Bean定義ファイル	
12	観ジョブ前処理	parentJobPreLogicList	org.springframework.beans.factory.config.ListFactoryBean	分割ジョブ実行時の観ジョブの前処理設定	sourceList	前処理として実行するクラスをリストに定義する。複数定義した場合は、上位に定義した前処理から実行される。	任意
13	観ジョブ後処理	parentJobPostLogicList	org.springframework.beans.factory.config.ListFactoryBean	分割ジョブ実行時の観ジョブの後処理の設定	sourceList	後処理として実行するクラスをリストに定義する。複数定義した場合は、上位に定義した後処理から実行される。	任意
14	分割キー取得	partitionKeyCollector	フレームワークBean定義ファイル説明書の分割キー処理用コレクタ定義から選択。	分割ジョブ実行時の分割キー取得処理定義	任意のプロパティ「ibatidsPartitionKeyCollector」を選択した場合は「sql」	「sql」の場合は、IBATISのSQL定義ファイルで定義されたSQL IDを設定する。	分割ジョブの場合は必須
15	対象データ取得	collector	任意のクラス名	対象データ取得用Beanの設定	任意のプロパティ		○
16	ビジネスロジック	bllogic	ビジネスロジックの実装クラス	ビジネスロジックのBeanの設定	任意のプロパティ		○
17	ジョブコンテキスト	jobContext	パラメータ値等を格納するクラス	パラメータ値等を格納するクラスの設定	任意のプロパティ		○
18	ジョブ前処理	jobPreLogicList	org.springframework.beans.factory.config.ListFactoryBean	ジョブ実行時のジョブの前処理設定	sourceList	前処理として実行するクラスをリストに定義する。複数定義した場合は、上位に定義した前処理から実行される。	任意
19	ジョブ後処理	jobPostLogicList	org.springframework.beans.factory.config.ListFactoryBean	ジョブ実行時のジョブの後処理設定	sourceList	後処理として実行するクラスをリストに定義する。複数定義した場合は、上位に定義した後処理から実行される。	任意
20	先頭チャンク前処理	firstChunkPreLogicList	org.springframework.beans.factory.config.ListFactoryBean	ジョブ実行時の先頭チャンク前処理設定	sourceList	前処理として実行するクラスをリストに定義する。複数定義した場合は、上位に定義した前処理から実行される。	任意
21	最終チャンク後処理	lastChunkPostLogicList	org.springframework.beans.factory.config.ListFactoryBean	ジョブ実行時の最終チャンク後処理設定	sourceList	後処理として実行するクラスをリストに定義する。複数定義した場合は、上位に定義した後処理から実行される。	任意
22	セーブポイント	useSavepoint	org.springframework.beans.factory.config.FieldRetrievingFactoryBean	セーブポイント使用有無設定	staticField	使用する場合は「java.lang.Boolean.TRUE」を設定	任意
23	チャンクサイズ	chunkSize	java.lang.Integer	チャンクサイズ設定	-		任意
24	分割ジョブの多重起動数	multiplicity	java.lang.Integer	分割ジョブの多重起動数設定	-		分割ジョブの場合は必須
25	SqlMapConfig	sqlMapConfigFileName	java.lang.String	SqlMapConfigの設定	任意のファイル	ジョブ毎にSqlMapを用意する場合に設定する。	任意
26	コントロールブレイク定義情報リスト	controlBreakDefinitionList	org.springframework.beans.factory.config.ListFactoryBean	コントロールブレイクのブレイク定義情報の設定	sourceList	コントロールブレイクの定義情報をリストに定義する。複数定義した場合は、上位から処理される。リストに定義する内容については項番27, 28を参照。	任意 (コントロールブレイクに対応した雛形をインポートする)
27				コントロールブレイクのブレイクキーおよびブレイクキーに対応するハンドラの設定	breakKey	コントロールブレイクのブレイクキーをリストに定義する。	任意 (コントロールブレイクに対応した雛形をインポートする)
28			jp.terasoluna.fw.batch.controlbreak.ControlBreakDefinitionItem		controlBreakHandler	ブレイクキーに対応したコントロールブレイクハンドラを定義する。	任意 (コントロールブレイクに対応した雛形をインポートする)
29	チャンクコントロールブレイク定義情報	chunkControlBreakDefinitionItem		チャンクコントロールブレイクのブレイクキーおよびブレイクキーに対応するハンドラの設定	breakKey	チャンクコントロールブレイクのブレイクキーをリストに定義する。	任意 (コントロールブレイクに対応した雛形をインポートする)
30			jp.terasoluna.fw.batch.controlbreak.ControlBreakDefinitionItem		controlBreakHandler	ブレイクキーに対応したコントロールブレイクハンドラを定義する。	任意 (コントロールブレイクに対応した雛形をインポートする)
31	トランスチャンクコントロールブレイク定義情報リスト	transControlBreakDefinitionList	org.springframework.beans.factory.config.ListFactoryBean	トランスチャンクコントロールブレイクのブレイク定義情報の設定	sourceList	トランスチャンクコントロールブレイクの定義情報をリストに定義する。複数定義した場合は、上位から処理される。リストに定義する内容については項番32, 33を参照。	任意 (コントロールブレイクに対応した雛形をインポートする)
32	トランスチャンクコントロールブレイク定義情報リスト			トランスチャンクコントロールブレイクのブレイクキーおよびブレイクキーに対応するハンドラの設定	breakKey	トランスチャンクコントロールブレイクのブレイクキーをリストに定義する。	任意 (コントロールブレイクに対応した雛形をインポートする)
33			jp.terasoluna.fw.batch.controlbreak.ControlBreakDefinitionItem		controlBreakHandler	ブレイクキーに対応したコントロールブレイクハンドラを定義する。	任意 (コントロールブレイクに対応した雛形をインポートする)

デフォルトBean定義ファイル

項番	設定項目名称	Bean ID	Classes/parent名	Bean概要	property	Property概要
1	ジョブ終了監視用クラス定義	endFileChecker	jp.terasoluna.fw.batch.init.EndFileChecker	ジョブ終了制御用Bean定義。	endFileDir	終了ファイル検出ディレクトリパス。初期設定値は『batchapps/BE-03/EndFile』である。
2		scheduledTask	org.springframework.scheduling.timer.ScheduledTimerTask	スケジュールタスク用Bean定義。 Common)対応機能を使用する際には『WorkManagerTaskContext-batch.xml』の設定が有効になるため『DefaultValueBean.xml』の設定は無効になる。	period timerTask	起動周期。 初期設定値は『8000』である。 スケジュール起動対象。 初期設定値は『endFileCheck』である。
3	タイマ	timerFactory	org.springframework.scheduling.timer.TimerFactoryBean	監視用タイマファクトリ用Bean定義。 Common)対応機能を使用する際には『WorkManagerTaskContext-batch.xml』の設定が有効になるため『DefaultValueBean.xml』の設定は無効になる。	daemon scheduledTimerTasks	初期設定値は『false』である。 スケジュールタスク用設定。 初期設定値は『scheduledTask』である。
4	デフォルト定義。ジョブBean定義に設定がある場合、ジョブBean定義が優先される。	useCache	org.springframework.beans.factory.config.FieldRerievingFactoryBean	BeanFactoryのキャッシュ有無のデフォルト設定。 ジョブBean定義に設定がある場合、ジョブBean定義が優先される。	staticField	初期設定値は『java.lang.Boolean.FALSE』である。
5		parentJobPreLogicList	org.springframework.beans.factory.config.ListFactoryBean	ジョブ分割前処理のデフォルトBean定義。 ジョブBean定義に設定がある場合、ジョブBean定義が優先される。	sourceList	初期設定値は『空白リスト』である。
6		parentJobPostLogicList	org.springframework.beans.factory.config.ListFactoryBean	ジョブ分割後処理のデフォルトBean定義。 ジョブBean定義に設定がある場合、ジョブBean定義が優先される。	sourceList	初期設定値は『空白リスト』である。
7		jobPreLogicList	org.springframework.beans.factory.config.ListFactoryBean	ジョブ前処理のデフォルトBean定義。 ジョブBean定義に設定がある場合、ジョブBean定義が優先される。	sourceList	初期設定値は『空白リスト』である。
8		jobPostLogicList	org.springframework.beans.factory.config.ListFactoryBean	ジョブ後処理のデフォルトBean定義。 ジョブBean定義に設定がある場合、ジョブBean定義が優先される。	sourceList	初期設定値は『空白リスト』である。
9		firstchunkPreLogicList	org.springframework.beans.factory.config.ListFactoryBean	先頭チャンク前処理のデフォルトBean定義。 ジョブBean定義に設定がある場合、ジョブBean定義が優先される。	sourceList	初期設定値は『空白リスト』である。
10		lastchunkPostLogicList	org.springframework.beans.factory.config.ListFactoryBean	最終チャンク後処理のデフォルトBean定義。 ジョブBean定義に設定がある場合、ジョブBean定義が優先される。	sourceList	初期設定値は『空白リスト』である。
11		chunkSize	java.lang.Integer	チャンクサイズ用Bean定義。 ジョブBean定義に設定がある場合、ジョブBean定義が優先される。	-	初期設定値は『20』である。
12		useSavepoint	org.springframework.beans.factory.config.FieldRerievingFactoryBean	セーブポイント有無のデフォルトBean定義。 ジョブBean定義に設定がある場合、ジョブBean定義が優先される。	staticField	初期設定値は『java.lang.Boolean.FALSE』である。
13		JobStatusSummarizer	jp.terasoluna.fw.batch.standard.StandardJobStatusSummarizer	処理状態の結果ハンドラのデフォルトBean定義。 ジョブBean定義に設定がある場合、ジョブBean定義が優先される。	-	-
14		exitCodeMap	org.springframework.beans.factory.config.FieldRerievingFactoryBean	終了コードのデフォルトBean定義。ジョブBean定義に設定がある場合、ジョブBean定義が優先される。	sourceMap	初期設定は以下の通り。 Key:ENDING_NORMALLY value:0 Key:ENDING_ABNORMALLY value:100 Key:SUSPENDING value:200
15		useMonitable	org.springframework.beans.factory.config.FieldRerievingFactoryBean	ジョブ監視有無のデフォルトBean定義。ジョブBean定義に設定がある場合、ジョブBean定義が優先される。	staticField	初期設定値は『java.lang.Boolean.FALSE』である。
16		manageableJobSize	java.lang.Integer	ジョブ監視の上限値用Bean定義。 ジョブBean定義に設定がある場合、ジョブBean定義が優先される。	-	初期設定値は『20』である。
17		sqlMapConfigFileName	java.lang.String	SqlMapConfigのデフォルトBean定義。ジョブBean定義に設定がある場合、ジョブBean定義が優先される。	sqlMapConfigFileName	初期設定値は『common/sql-map-config.xml』である。
18		chunkQueueLength	java.lang.Integer	チャンクキューのサイズ用Bean定義。 ジョブBean定義に設定がある場合、ジョブBean定義が優先される。	-	初期設定値は『10』である。
19		partitionQueueLength	java.lang.Integer	分割キューのサイズ用Bean定義。 ジョブBean定義に設定がある場合、ジョブBean定義が優先される。	-	初期設定値は『10』である。
20	例外ハンドラ	exceptionHandlerMap	org.springframework.beans.factory.config.MapFactoryBean	例外ハンドラマップのBean定義。	sourceMap	ハンドラマップ値の設定。 初期設定値は『CollectorException=jp.terasoluna.batch.sample.bc01.CollectorExceptionHandlerImpl JobException=jp.terasoluna.fw.batch.standard.StandardJobExceptionHandler』である。
21	デフォルト例外ハンドラ定義	defaultJobExceptionHandler	jp.terasoluna.fw.batch.standard.StandardJobExceptionHandler	JobException用デフォルト例外ハンドラのBean定義。	なし	
22	Throwableハンドラ定義	throwableHandler	jp.terasoluna.fw.batch.standard.StandardThrowableHandler	ThrowableハンドラのBean定義	なし	
23	処理結果ハンドラ	BLogicResultHandler	jp.terasoluna.fw.batch.standard.StandardBLogicResultHandler	ビジネスロジック結果ハンドラBean定義。 BLogicResultHandlerの実装クラス。	なし	
24		TransactionalBLogicResultHandler	jp.terasoluna.fw.batch.springsupport.transaction.TransactionalBLogicResultHandler	チャック別トランザクションモデル用ビジネスロジック結果ハンドラBean定義。 BLogicResultHandlerの拡張クラス。	なし	
25		CollectorResultHandler	jp.terasoluna.fw.batch.standard.StandardCollectorResultHandler	Collector結果ハンドラBean定義。 StandardBLogicResultHandlerの実装クラス。	なし	
26		batchUpdateResultHandler	jp.terasoluna.fw.batch.standard.StandardBatchUpdateResultHandler	バッチ更新処理結果ハンドラBean定義。 BatchUpdateResultHandlerの実装クラス。	なし	
27		supportLogicResultHandler	jp.terasoluna.fw.batch.standard.StandardSupportLogicResultHandler	サポート処理実行結果ハンドラBean定義。 SupportLogicResultHandlerの実装クラス。	なし	
28	入力チェック	validationConfigLocations	org.springframework.beans.factory.config.ListFactoryBean	入力チェック定義ファイルとルールファイルのBean定義。	sourceList	初期値は『/common/validator-rules.xml、 /common/validator-rules-ex.xml、 /common/validationCommon.xml』である。
29		validationResultHandler	jp.terasoluna.fw.batch.validation.StandardValidationResultHandler	入力チェック処理結果ハンドラBean定義。 ValidationResultHandlerの実装クラス。	なし	

データアクセスBean定義ファイル

項番	設定項目名称	Bean ID	Classes/parent名	Bean概要	property	Property概要
1	データソース	dataSource	org.apache.commons.dbcp.BasicDataSource等	DataSourceのBean定義。 DBCPによるコネクションプーリングを行う。	driverClassName	JDBCのドライバの設定。
					url	接続先のURLの設定。
					username	ユーザアカウントの設定。
					password	パスワードの設定。
					poolPreparedStatements	PreparedStatementsのプーリング有無（デフォルトはtrue）
					defaultAutoCommit	オートコミットの有無（デフォルトはfalse）
2	トランザクションマネージャ	transactionManager	org.springframework.jdbc.datasource.DataSourceTransactionManager	単一のJDBCデータソース向けのトランザクションマネージャBean定義。	dataSource	使用するデータソースBeanの設定。 初期設定値は『dataSource』である。
3	トランザクションプロキシ	baseTransactionProxy	org.springframework.transaction.interceptor.TransactionProxyFactoryBean	トランザクションプロキシのベース定義となる親Bean定義。 Worker定義のtarget属性に指定するBean。子のBean定義で属性をオーバーライドすることができる。 Springで提供されているファクトリクラスのBean定義。 任意のクラスにトランザクション制御コードを付加したプロキシクラスを生成する。	transactionManager	トランザクションマネージャの設定。 プロキシ対象のBean、およびトランザクション定義情報の設定は、このbaseTransactionProxyを親定義とするフレームワークのBean定義で行われている。 初期設定値は『transactionManager』である。
4	SQLMapClient	sqlMapClient	org.springframework.orm.ibatis.SqlMapClientFactoryBean	iBATIS データベース層のためのSQLMapのBean定義。	configLocation	使用するSQLMapの定義ファイルの設定。 初期設定値は『sqlMapConfigFileName』である。（デフォルトBean定義ファイル参照）
					dataSource	使用するデータソースBeanの設定。 初期設定値は『dataSource』である。
5	DAO	queryDAO	jp.terasoluna.fw.dao.ibatis.QueryDAOImpl	参照系のDAOのBean定義。	sqlMapClient	使用するSQLMapClientの設定。 初期設定値は『sqlMapClient』である。
6		updateDAO	jp.terasoluna.fw.dao.ibatis.UpdateDAOImpl	更新系のDAOのBean定義。	sqlMapClient	使用するSQLMapClientの設定。 初期設定値は『sqlMapClient』である。
7		spDAO	jp.terasoluna.fw.dao.ibatis.StoredProcedureDAOImpl	ストアドプロシジャ用のDAOのBean定義。	sqlMapClient	使用するSQLMapClientの設定。 初期設定値は『sqlMapClient』である。



非同期型ジョブ起動機能パッチデモン

項番	設定項目名称	Bean ID	Classes/parent名	概要	property	property概要
1	非同期パッチ起動用マネージャ	jobManager	baseManager (jp.terasoluna.fw.batch.core.JobManager)	非同期パッチ起動用マネージャの定義。	name collector workQueueFactory preProcessor postProcessor	初期値『AsyncBatchDaemonManager』 項番16を参照 項番2を参照 項番3を参照 項番5を参照
2		-	jp.terasoluna.fw.batch.standard.StandardWorkQueueFactory	非同期パッチ起動用作業キュー作成クラスの定義。	workerCircularService queueProcessor queueLength multiplicity	フレームワークBean定義ファイルを参照 項番5を参照 初期値『10』 項番29を参照
3		parentJobPreProcessor	transactionalSupportProcessor (jp.terasoluna.fw.batch.springsupport.transaction.TransactionalSupportProcessor)	トランザクション管理有りのジョブ前処理の定義。	supportProcessor	項番4を参照
4		-	standardSupportProcessor (jp.terasoluna.fw.batch.standard.StandardSupportProcessor)	ジョブ前処理の定義。	-	-
5		parentJobPostProcessor	transactionalSupportProcessor (jp.terasoluna.fw.batch.springsupport.transaction.TransactionalSupportProcessor)	トランザクション管理有りのジョブ後処理の定義。	supportProcessor	項番6を参照
6		-	standardSupportProcessor (jp.terasoluna.fw.batch.standard.StandardSupportProcessor)	ジョブ後処理の定義。	-	-
7		jobContext	jp.terasoluna.fw.batch.init.RootJobContext	親ジョブ用ジョブコンテキストの定義。	-	-
8	キュープロセッサ	asyncBatchDaemonQueueProcessor	baseQueueProcessor (jp.terasoluna.fw.batch.standard.QueueProcessor)	親ジョブ用キュー処理クラスの定義。	name worker constructor-arg	初期値『asyncBatchDaemonQueueProcessor』 項番9を参照 Bean定義ファイル 初期値 『/common/FrameworkBean.xml,/common/dataAccessContext-batch.xml,/common/ThreadPoolContext-AsyncBatch.xml』
9	パッチ起動用ワーク	childJobManager	jp.terasoluna.fw.batch.springsupport.init.JobExecutor	ジョブ起動用クラスの定義。	async preJobProcessor postJobProcessor throwableHandler	非同期起動有無フラグ 初期値『true』 項番10を参照 項番13を参照 ThrowableハンドラのBean定義。 初期設定値は『throwableHandler』である。 デフォルトBean定義ファイルを参照。
10		preJobProcessor	transactionalSupportProcessor (jp.terasoluna.fw.batch.springsupport.transaction.TransactionalSupportProcessor)	トランザクション管理有りのジョブ前処理の定義。	supportProcessor	項番11を参照
11		-	standardSupportProcessor (jp.terasoluna.fw.batch.standard.StandardSupportProcessor)	ジョブ前処理の定義。	supportLogicList	項番12を参照
12		-	jp.terasoluna.fw.batch.springsupport.init.AsyncJobPreLogic	非同期用ジョブ前処理の定義。	jobControlTableHandler	フレームワークBean定義ファイルを参照
13		postJobProcessor	transactionalSupportProcessor (jp.terasoluna.fw.batch.springsupport.transaction.TransactionalSupportProcessor)	トランザクション管理有りのジョブ後処理の定義。	supportProcessor	項番14を参照
14		-	standardSupportProcessor (jp.terasoluna.fw.batch.standard.StandardSupportProcessor)	ジョブ後処理の定義。	supportLogicList	項番15を参照
15		-	jp.terasoluna.fw.batch.springsupport.init.AsyncJobPreLogic	非同期用ジョブ後処理の定義。	jobControlTableHandler	フレームワークBean定義ファイルを参照
16	コレクター	batchDaemonJobCollector	jp.terasoluna.fw.batch.init.JobRequestInfoCollector	非同期ジョブ起動時のジョブ依頼情報取得用クラスの定義。	collectedDataHandlerFactory jobControlTableHandler intervalSeconds	項番17を参照 フレームワークBean定義ファイルを参照 ジョブ管理テーブルの監視周期(秒) 初期値『10』
17		-	jp.terasoluna.fw.batch.init.JobRequestInfoHandlerFactory	ジョブ依頼情報格納用ワークキュー生成用クラスの定義。	-	-
18	ジョブの処理状況クラス定義	MonitorableJobStatus	jp.terasoluna.fw.batch.monitor.MonitorableJobStatus	監視用ジョブの処理状況用Bean定義。	mbeanRegister transactionManager useSavepoint	ジョブ監視用Bean定義 初期設定値は『MBeanRegister』であり、監視を行わない場合は設定する必要はない。 トランザクションマネージャのBean定義。 データアクセスBean定義ファイルを参照。 セーブポイント有無のBean定義。 デフォルトBean定義ファイルを参照。
19		-	-	-	partitionNo	ジョブパーティション番号のBean定義。 初期設定値は『-1』である。
20		-	-	-	transactionManager	トランザクションマネージャのBean定義。 データアクセスBean定義ファイルを参照。
21		-	-	-	useSavepoint	セーブポイント有無のBean定義。 デフォルトBean定義ファイルを参照。
22	ジョブの処理状況クラス	JobStatus	jp.terasoluna.fw.batch.springsupport.transaction.TransactionalJobStatus	ジョブの処理状況用Bean定義。	transactionManager useSavepoint partitionNo	トランザクションマネージャのBean定義。 データアクセスBean定義ファイルを参照。 セーブポイント有無のBean定義。 デフォルトBean定義ファイルを参照。 ジョブパーティション番号のBean定義。 初期設定値は『-1』である。
23		-	-	-	-	-
24		-	-	-	-	-
25	ジョブ監視	useMonitorable	org.springframework.beans.factory.config.FieldRetrievingFactoryBean	ジョブ監視有無のBean定義。	staticField	初期値『java.lang.Boolean.FALSE』
26	ジョブ管理テーブル	jobControlTableHandler	jp.terasoluna.fw.batch.init.JobControlTableHandlerImpl	ジョブ管理テーブル用ハンドラのBean定義。	queryDAO updateDAO	参照系のDAOのBean定義。 データアクセスBean定義ファイルを参照。 更新系のDAOのBean定義。 データアクセスBean定義ファイルを参照。
27		-	-	-	-	-
28	スレッド管理	threadSize	jp.terasoluna.fw.batch.springsupport.standard.ThreadSizeFactoryBean	スレッドプール作成時のサイズ	multiplicity	項番29を参照。
29		multiplicity	java.lang.Integer	多重度	constructor-arg	初期値『1』

デーモン終了用ジョブBean定義ファイル

項番	設定項目名称	Bean ID	Classes/parent名	概要	property	property概要
1	非同期バッチ起動用のマネージャ	jobManager	jp.terasoluna.fw.batch.init.NullWorker	非同期バッチ起動用のマネージャ	-	-
2		jobContext	jp.terasoluna.fw.batch.spring.support.init.AsyncJobContext	ジョブコンテキスト	-	-
3	ジョブの処理状況クラス	MonitorableJobStatus	jp.terasoluna.fw.batch.monitor.MonitorableJobStatus	監視用ジョブの処理状況用Bean定義。	mbeanRegister	ジョブ監視用Bean定義 初期設定値は『MBeanRegister』であり、監視を行わない場合は設定する必要はない。
4					partitionNo	ジョブパーティション番号のBean定義。 初期設定値は『-1』である。
5		JobStatus	jp.terasoluna.fw.batch.core.JobStatus	ジョブの処理状況用Bean定義。	partitionNo	ジョブパーティション番号のBean定義。 初期設定値は『-1』である。

非同期型起動用スレッドプール定義ファイル

順番	設定項目名称	Bean ID	Classes/parent名	Bean概要	property	Property概要
1	スレッドプール管理	workerExecutorService	jp.terasoluna.fw.batch.springsupport.standard.FixedThreadPoolWorkerExecutorService	フレームワークが利用するスレッドプールのBean定義。	constructor-arg	スレッド数の設定。 初期設定値は『20』である。
					defaultJobExceptionHandler	デフォルト例外ハンドラの設定。 初期設定値は『defaultJobExceptionHandler』である。
					exceptionHandlerMap	例外ハンドラマップの設定。 初期設定値は『exceptionHandlerMap』である。
					throwableHandler	ThrowableハンドラのBean定義。 初期設定値は『throwableHandler』である。 デフォルトBean定義ファイルを参照。

同期型起動用スレッドプール定義ファイル

項目	設定項目名称	Bean ID	Classes/parent名	Bean概要	property	Property概要
1	スレッドプール管理	workerExecutorService	jp.terasoluna.fw.batch.spring.support.standard.FixedThreadPoolWorkerExecutorService	フレームワークが利用するスレッドプールのBean定義。	constructor-arg	スレッド数の設定。 初期設定値は『threadSize』である。
					defaultJobExceptionHandler	デフォルト例外ハンドラの設定。 初期設定値は『defaultJobExceptionHandler』である。
					exceptionHandlerMap	例外ハンドラマップの設定。 初期設定値は『exceptionHandlerMap』である。
					throwableHandler	ThrowableハンドラのBean定義。 初期設定値は『throwableHandler』である。
						デフォルトBean定義ファイルを参照。

## 非同期型ジョブ起動機能バッチデーモン(CommonJ用)

順番	設定項目名称	Bean ID	Classes/parent名	概要	property	property概要
1	非同期バッチ起動用マネージャ	jobManager	baseManager (jp.terasoluna.fw.batch.core.JobManager)	非同期バッチ起動用マネージャの定義。	name collector workQueueFactory preProcessor postProcessor	初期値『AsyncBatchDaemonManager』 項番16を参照 項番2を参照 項番3を参照 項番5を参照
2		-	jp.terasoluna.fw.batch.common.J.WorkManagerTaskWorkQueueFactory	非同期バッチ起動用作業キュー作成クラスの定義(CommonJ用)。	workerCalculatorService queueProcessor queueLength multiplicity	フレームワークBean定義ファイルを参照 項番5を参照 初期値『10』 項番29を参照
3		parentJobPreProcessor	transactionalSupportProcessor (jp.terasoluna.fw.batch.springsupport.transactionalSupportProcessor)	トランザクション管理有りのジョブ前処理の定義。	supportProcessor	項番4を参照
4		-	standardSupportProcessor (jp.terasoluna.fw.batch.standard.StandardSupportProcessor)	ジョブ前処理の定義。	-	-
5		parentJobPostProcessor	transactionalSupportProcessor (jp.terasoluna.fw.batch.springsupport.transactionalSupportProcessor)	トランザクション管理有りのジョブ後処理の定義。	supportProcessor	項番6を参照
6		-	standardSupportProcessor (jp.terasoluna.fw.batch.standard.StandardSupportProcessor)	ジョブ後処理の定義。	-	-
7		jobContext	jp.terasoluna.fw.batch.init.RootJobContext	親ジョブ用ジョブコンテキストの定義。	-	-
8	キュープロセッサ	asyncBatchDaemonQueueProcessor	baseQueueProcessor (jp.terasoluna.fw.batch.standard.QueueProcessor)	親ジョブ用キュー処理クラスの定義。	name	初期値『asyncBatchDaemonQueueProcessor』
9	バッチ起動用ワーク	childJobManager	jp.terasoluna.fw.batch.common.J.init.JobExecutor	ジョブ起動用クラスの定義。	worker constructor-arg async preJobProcessor postJobProcessor jobParametersSplitStr throwableHandler	初期値 Bean定義ファイル 初期値 『/common/FrameworkBean.xml,/common/dataAccessContext- -batch.xml,/common/WorkManagerTaskContext- -batch.xml,/common/ValidationContext-AsyncBatch.xml』 非同期起動有無フラグ 初期値『true』 項番10を参照 項番13を参照 ジョブパラメータの区切り文字 初期設定値は『,』である。 ThrowableハンドラのBean定義。 初期設定値は『throwableHandler』である。 デフォルトBean定義ファイルを参照。
10			transactionalSupportProcessor (jp.terasoluna.fw.batch.springsupport.transactionalSupportProcessor)	トランザクション管理有りのジョブ前処理の定義。	supportProcessor	項番11を参照
11			standardSupportProcessor (jp.terasoluna.fw.batch.standard.StandardSupportProcessor)	ジョブ前処理の定義。	supportLogicList	項番12を参照
12			jp.terasoluna.fw.batch.springsupport.init.AsyncJobPreLogic	非同期用ジョブ前処理の定義。	jobControlTableHandler	フレームワークBean定義ファイルを参照
13			transactionalSupportProcessor (jp.terasoluna.fw.batch.springsupport.transactionalSupportProcessor)	トランザクション管理有りのジョブ後処理の定義。	supportProcessor	項番14を参照
14			standardSupportProcessor (jp.terasoluna.fw.batch.standard.StandardSupportProcessor)	ジョブ後処理の定義。	supportLogicList	項番15を参照
15			jp.terasoluna.fw.batch.springsupport.init.AsyncJobPostLogic	非同期用ジョブ後処理の定義。	jobControlTableHandler	フレームワークBean定義ファイルを参照
16	コレクター	batchDaemonJobCollector	jp.terasoluna.fw.batch.init.JobRequestInfoCollector	非同期ジョブ起動時のジョブ依頼情報取得用クラスの定義。	collectedDataHandlerFactory jobControlTableHandler	項番17を参照 フレームワークBean定義ファイルを参照
17		-	jp.terasoluna.fw.batch.init.JobRequestInfoHandlerFactory	ジョブ依頼情報格納用ワークキュー生成用クラスの定義。	intervalSeconds refreshCount	ジョブ管理テーブルの監視周期(秒) 初期値『10』 初期値『20』
18	ジョブの処理状況クラス定義	MonitorableJobStatus	jp.terasoluna.fw.batch.monitor.MonitorableTransactionalJobStatus	監視用ジョブの処理状況用Bean定義。	mbeanRegister	ジョブ監視用Bean定義 初期設定値は『MBeanRegister』であり、監視を行わない場合は設定する必要はない。
19		ジョブの処理状況クラス	JobStatus	ジョブの処理状況用Bean定義。	transactionManager	トランザクションマネージャのBean定義。 データアクセスBean定義ファイルを参照。
20					useSavepoint	セーブポイント有無のBean定義。 デフォルトBean定義ファイルを参照。
21	ジョブ監視	useMonitorable	org.springframework.beans.factory.config.FieldRetrievingFactoryBean	ジョブ監視有無のBean定義。	partitionNo	ジョブパーティション番号のBean定義。 初期設定値は『-1』である。
22					transactionManager	トランザクションマネージャのBean定義。 データアクセスBean定義ファイルを参照。
23					useSavepoint	セーブポイント有無のBean定義。 デフォルトBean定義ファイルを参照。
24	ジョブ管理テーブル	jobControlTableHandler	jp.terasoluna.fw.batch.init.JobControlTableHandlerImpl	ジョブ管理テーブル用ハンドラのBean定義。	partitionNo	ジョブパーティション番号のBean定義。 初期設定値は『-1』である。
25					staticField	初期値『java.lang.Boolean.FALSE』
26	ジョブ管理テーブル	jobControlTableHandler	jp.terasoluna.fw.batch.init.JobControlTableHandlerImpl	ジョブ管理テーブル用ハンドラのBean定義。	queryDAO	参照系のDAOのBean定義。 データアクセスBean定義ファイルを参照。
27					updateDAO	更新系のDAOのBean定義。 データアクセスBean定義ファイルを参照。
28	スレッド管理	threadSize	jp.terasoluna.fw.batch.springsupport.standard.ThreadSizeFactoryBean	スレッドプール作成時のサイズ	multiplicity	項番29を参照。
29		multiplicity	java.lang.Integer	多重度	constructor-arg	初期値『3』

ジョブスターター用ワークマネージャ定義ファイル

項番	設定項目名称	Bean ID	Classes/parent名	概要	property	property概要
1	ジョブスターター	jobStarter	jp.terasoluna.fw.batch.common.init.JobStarterImpl	ジョブ起動の定義	workManager workListener	項番2を参照。 初期値は『manager』である。 項番3を参照。 初期値は『listener』である。
2	ワークマネージャ	manager	org.springframework.scheduling.common.WorkManagerTaskExecutor	ワークマネージャの定義	workManagerName resourceRef	weblogic-application.xml に記述される、WorkManager 定義のname属性を参照。 初期値は『wm/BatchWorkManager』である。 初期値は『true』である。
3	ワークリスナー	listener	jp.terasoluna.fw.batch.common.listener.JobExecutorListener	ワークリスナーの定義	jobResultInfoHandler	項番6を参照。
4	SqlMapConfigファイルPath	sqlMapConfigFileName	java.lang.String	SqlMapConfigファイルのパス定義	constructor-arg	使用するSqlMapConfigの定義ファイルの設定。 初期値は『common/sql-map-config.xml』である。
5	ジョブ管理テーブル用ハンドラ	jobMessageTableHandler	jp.terasoluna.fw.batch.jobmessage.JobMessageTableHandlerImpl	ジョブ管理テーブル用ハンドラの定義	queryDAO updateDAO	参照系のDAOのBean定義。 初期値は『queryDAO』である。 更新系のDAOのBean定義。 初期値は『updateDAO』である。
6	ジョブ結果テーブル用ハンドラ	jobResultInfoHandler	jp.terasoluna.fw.batch.common.transaction.JobResultInfoHandlerImpl	ジョブ結果テーブル用ハンドラの定義	updateDAO transactionManager	参照系のDAOのBean定義。 初期値は『updateDAO』である。 データアクセスBean定義を参照。 初期値は『transactionManager』である。

ワーク用ワークマネージャ定義ファイル

順番	設定項目名称	Bean ID	Classes/parent名	概要	property	property概要
1	ワーク管理	workExecutorService	jp.terasoluna.fw.batch.common.WorkManagerTaskExecutorService	ワーク管理の定義	defaultJobExceptionHandler exceptionHandlerMap workManager workListener throwableHandler	デフォルトBean定義ファイルを参照。 初期値は『defaultJobExceptionHandler』である。 デフォルトBean定義ファイルを参照。 初期値は『exceptionHandlerMap』である。 項番2を参照。 初期値は『TaskExecutor』である。 項番3を参照。 初期値は『workListener』である。 ThrowableハンドラのBean定義。 初期設定値は『throwableHandler』である。 デフォルトBean定義ファイルを参照。
2		taskExecutor	org.springframework.scheduling.common.WorkManagerTaskExecutor	タスク実行の定義	workManagerName resourceRef	weblogic-application.xmlに記述される、WorkManager定義のname属性を参照。 初期値は『wm.BatchWorkManager』である。 初期値は『true』である。
3		workListener	jp.terasoluna.fw.batch.common.listener.WorkQueueListener	ワークリスナーの定義	-	-
4	監視	scheduledTask	org.springframework.scheduling.common.ScheduledTimerListener	スケジューラの定義 Common対応機能を使用する際には『WorkManagerTaskContext-batch.xml』の設定が有効になるため『DefaultValueBean.xml』の設定は無効になる。	delay period fixedRate runnable	遅延時間の設定。 初期値は『0』である。 繰り返す周期の設定。 初期値は『8000』である。（単位は『ms』） 初期値は『false』である。 デフォルトBean定義ファイルを参照。 初期値は『endFileChecker』である。
5		timerFactory	org.springframework.scheduling.common.TimerManagerFactoryBean	タイマーの定義 Common対応機能を使用する際には『WorkManagerTaskContext-batch.xml』の設定が有効になるため『DefaultValueBean.xml』の設定は無効になる。	timerManagerName resourceRef shared scheduledTimerListeners jndiTemplate	weblogic-application.xmlに記述される、WorkManager定義のname属性を参照。 初期値は『timer/FileCheckTimer』である。 初期値は『true』である。 初期値は『true』である。 タイマーリスナーのBean定義。 listタグにより複数のリスナーを設定できる。 初期値は『scheduledTask』である。 JNDI環境設定。 初期値は『jndiTemplateForTimerManager』である。

ブレースフォルダ設定定義ファイル

項番	設定項目名称	Bean ID	Classes/parent名	概要	property	property概要
1	ブレースホルダ	-	org.springframework.beans.factory.config.PropertyPlaceholderConfigurer	ジョブが使用するブレースホルダの設定定義。	locations	使用するブレースホルダのパスを定義する。 初期値は『jdbc.properties』 『template/workQueueFactory.properties』



チャンク別トランザクションモデル

順番	設定項目名称	Bean ID	Classes/parent名	概要	property	property概要
0	ブレースホルダ			ブレースホルダの定義	import resource	初期値: [classpath:common/PlaceholderConfig.xml]
1	マネージャ	jobManager	baseManager (jp.terasoluna.fw.batch.core.JobManager)	ジョブ実行マネージャの定義。	name collector workQueueFactory preProcessor postProcessor	初期値: [chunkTransactionJobManager] ジョブBean定義ファイルを参照 項番2を参照 項番3を参照 項番5を参照
2		-	jp.terasoluna.fw.batch.commonj.WorkManagerTaskWorkerQueueFactory	commonj対応作業キュー生成クラスの定義。	workerExecuterService queueLength queueProcessor	同期型起動用スレッドプール定義ファイルを参照 非同期型起動用スレッドプール定義ファイルを参照 デフォルトBean定義ファイルを参照 項番7を参照
3		jobPreProcessor	transactionalSupportProcessor (jp.terasoluna.fw.batch.spring.support.transactionalSupportProcessor)	トランザクション制御有りのジョブ前処理の定義。	supportProcessor	項番4を参照
4		-	standardSupportProcessor (jp.terasoluna.fw.batch.standard.StandardSupportProcessor)	ジョブ前処理の定義。	supportLogicList	デフォルトBean定義ファイルを参照 ジョブBean定義ファイルを参照
5		jobPostProcessor	transactionalSupportProcessor (jp.terasoluna.fw.batch.spring.support.transactionalSupportProcessor)	トランザクション制御有りのジョブ後処理の定義。	supportProcessor	項番6を参照
6		-	standardSupportProcessor (jp.terasoluna.fw.batch.standard.StandardSupportProcessor)	ジョブ後処理の定義。	supportLogicList	デフォルトBean定義ファイルを参照 ジョブBean定義ファイルを参照
7	キュープロセッサ	chunkTransactionQueueProcessor	baseQueueProcessor (jp.terasoluna.fw.batch.standard.QueueProcessor)	キュー処理クラスの定義。	name worker	初期値: [chunkTransactionQueueProcessor] 項番8を参照
8	ワーカ	chunkTransactionWorker	jp.terasoluna.fw.batch.spring.support.transactionalTransactionWorker jp.terasoluna.fw.batch.core.JobWorker	トランザクション制御有りのワーカ定義。 チャンク処理クラスの定義。	jobWorker	項番9を参照
9		jobWorker			biologicExecutor batchUpdateProcessor exceptionHandlerMap batchUpdateResultHandler defaultJobExceptionHandler throwableHandler	項番10を参照 項番11を参照 デフォルトBean定義ファイルを参照 デフォルトBean定義ファイルを参照 デフォルトBean定義ファイルを参照 ThrowableハンドラのBean定義。 初期設定値は [throwableHandler] である。 デフォルトBean定義ファイルを参照。
10		biologicExecutor	jp.terasoluna.fw.batch.core.StandardBiologicExecutor	ビジネスロジック実行クラスの定義。	biologic biologicResultHandler exceptionHandlerMap defaultJobExceptionHandler throwableHandler	ジョブBean定義ファイルを参照 デフォルトBean定義ファイルを参照 デフォルトBean定義ファイルを参照 デフォルトBean定義ファイルを参照 ThrowableハンドラのBean定義。 初期設定値は [throwableHandler] である。 デフォルトBean定義ファイルを参照。
11		batchUpdateProcessor	jp.terasoluna.fw.batch.standard.StandardBatchUpdateProcessor	バッチ更新処理実行クラスの定義。	updateDAO	データアクセスBean定義ファイルを参照
12	ジョブの処理状況クラス	MonitorableJobStatus	jp.terasoluna.fw.batch.monitor.MonitorableJobStatus	監視用ジョブの処理状況用Bean定義。	mbeanRegister transactionManager useSavepoint partitionNo	フレームワークBean定義ファイルを参照 データアクセスBean定義ファイルを参照 デフォルトBean定義ファイルを参照 初期値: [-1]
13		JobStatus	jp.terasoluna.fw.batch.core.JobStatus	ジョブの処理状況用Bean定義。	partitionNo transactionManager useSavepoint	初期値: [-1] データアクセスBean定義ファイルを参照 デフォルトBean定義ファイルを参照
14	スレッドプールサイズ	threadSize	java.lang.Integer	スレッドプールのサイズ	constructor-arg	初期値: [2]

チャック別トランザクションモデル (リスタート)

項番	設定項目名称	Bean ID	Classes/parent名	概要	property	property概要
0	ブレースホルダ			ブレースホルダの定義	import resource	初期値 [classpath:common/PlaceholderConfig.xml]
1	マネージャ	jobManager	jp.terasoluna.fw.batch.restart.JobContextRestorer	リスタート実行用マネージャ定義。	jobRestartTableHandler jobManager throwableHandler	初期値 [classpath:common/PlaceholderConfig.xml] 項番21を参照 項番2を参照 ThrowableハンドラのBean定義。 初期設定値は『throwableHandler』である。 デフォルトBean定義ファイルを参照。
2		delegatedJobManager	baseManager (jp.terasoluna.fw.batch.core.JobManager)	ジョブ実行マネージャの定義。	name collector workQueueFactory preProcessor postProcessor	初期値 [chunkTransactionJobManager] ジョブBean定義ファイルを参照 項番3を参照 項番4を参照 項番7を参照
3		-	jp.terasoluna.fw.batch.common.J.WorkManagerTaskWorkerQueueFactory	common)対応作業キュー生成クラスの定義。	workerExecutorService	同期型起動用スレッドプール定義ファイルを参照 非同期型起動用スレッドプール定義ファイルを参照
4		jobPreProcessor	jp.terasoluna.fw.batch.restart.RestartPreProcessor	リスタート時のジョブ前処理の定義。	queueLength queueProcessor jobRestartTableHandler preProcessor throwableHandler	デフォルトBean定義ファイルを参照 項番10を参照 項番21を参照 項番5を参照 ThrowableハンドラのBean定義。 初期設定値は『throwableHandler』である。 デフォルトBean定義ファイルを参照。
5		restartJobPreProcessor	transactionalSupportProcessor (jp.terasoluna.fw.batch.spring.support.transactionalSupportProcessor)	トランザクション制御有りのジョブ前処理の定義。	supportProcessor	項番6を参照
6		-	standardSupportProcessor (jp.terasoluna.fw.batch.standard.StandardSupportProcessor)	ジョブ前処理の定義。	supportLogicList	デフォルトBean定義ファイルを参照 ジョブBean定義ファイルを参照
7		jobPostProcessor	jp.terasoluna.fw.batch.restart.RestartPostProcessor	リスタート時のジョブ後処理の定義。	jobRestartTableHandler postProcessor throwableHandler	項番21を参照 項番8を参照 ThrowableハンドラのBean定義。 初期設定値は『throwableHandler』である。 デフォルトBean定義ファイルを参照。
8		delegatedJobPostProcessor	transactionalSupportProcessor (jp.terasoluna.fw.batch.spring.support.transactionalSupportProcessor)	トランザクション制御有りのジョブ後処理の定義。	supportProcessor	項番9を参照
9		-	standardSupportProcessor (jp.terasoluna.fw.batch.standard.StandardSupportProcessor)	ジョブ後処理の定義。	supportLogicList	デフォルトBean定義ファイルを参照 ジョブBean定義ファイルを参照
10	キュープロセッサ	chunkTransactionQueueProcessor	baseQueueProcessor (jp.terasoluna.fw.batch.standard.QueueProcessor)	キュー処理クラスの定義。	name worker	初期値 [chunkTransactionQueueProcessor] 項番11を参照
11	ワーカ	chunkTransactionWorker	jp.terasoluna.fw.batch.spring.support.transactional.TransactionalWorker	チャック処理クラスの定義。	transactionManager useSavepoint jobWorker jobRestartTableHandler throwableHandler	データベースBean定義ファイルを参照 デフォルトBean定義ファイルを参照 ジョブBean定義ファイルを参照 項番12を参照 項番14を参照 ThrowableハンドラのBean定義。 初期設定値は『throwableHandler』である。 デフォルトBean定義ファイルを参照。
12		restartPointUpdater	jp.terasoluna.fw.batch.restart.RestartPointUpdater	リスタートポイント登録用ワーククラスの定義。	jobWorker jobRestartTableHandler throwableHandler	項番14を参照 項番21を参照 ThrowableハンドラのBean定義。 初期設定値は『throwableHandler』である。 デフォルトBean定義ファイルを参照。
13		restartPointCloser	jp.terasoluna.fw.batch.restart.RestartPointCloser	リスタートポイント削除用ワーククラスの定義。	jobWorker jobRestartTableHandler supportLogicList sqlKey throwableHandler	項番14を参照 項番21を参照 デフォルトBean定義ファイルを参照 ジョブBean定義ファイルを参照 リスタート情報クリア用SQL 初期設定値は『jobRestart.DELETE_JOB_RESTART_CLEAR』である。 ThrowableハンドラのBean定義。 初期設定値は『throwableHandler』である。 デフォルトBean定義ファイルを参照。
14		jobWorker	jp.terasoluna.fw.batch.core.JobWorker	チャック処理クラスの定義。	biologicExecutor batchUpdateProcessor exceptionHandlerMap batchUpdateResultHandler defaultJobExceptionHandler throwableHandler	項番15を参照 項番16を参照 デフォルトBean定義ファイルを参照 デフォルトBean定義ファイルを参照 デフォルトBean定義ファイルを参照 ThrowableハンドラのBean定義。 初期設定値は『throwableHandler』である。 デフォルトBean定義ファイルを参照。
15		biologicExecutor	jp.terasoluna.fw.batch.core.StandardBiologicExecutor	ビジネスロジック実行クラスの定義。	biologic biologicResultHandler exceptionHandlerMap defaultJobExceptionHandler throwableHandler	ジョブBean定義ファイルを参照 デフォルトBean定義ファイルを参照 デフォルトBean定義ファイルを参照 デフォルトBean定義ファイルを参照 ThrowableハンドラのBean定義。 初期設定値は『throwableHandler』である。 デフォルトBean定義ファイルを参照。
16		batchUpdateProcessor	jp.terasoluna.fw.batch.standard.StandardBatchUpdateProcessor	バッチ更新処理実行クラスの定義。	updateDAO	データベースBean定義ファイルを参照
17		ジョブの処理状況クラス	MonitorableJobStatus	監視用ジョブの処理状況用Bean定義。	mbeanRegister transactionManager useSavepoint partitionNo defaultJobExitCodeMap	フレームワークBean定義ファイルを参照 データベースBean定義ファイルを参照 デフォルトBean定義ファイルを参照 初期値『1』 デフォルトBean定義ファイルを参照
18		JobStatus	jp.terasoluna.fw.batch.core.JobStatus	ジョブの処理状況用Bean定義。	partitionNo transactionManager useSavepoint partitionNo defaultJobExitCodeMap	初期値『1』 データベースBean定義ファイルを参照 デフォルトBean定義ファイルを参照 初期値『1』 デフォルトBean定義ファイルを参照
19	スレッドプールサイズ	threadSize	java.lang.Integer	スレッドプールのサイズ	constructor-arg	初期値『1』
20	ジョブリスタート管理テーブルハンドラ	JobRestartTableHandler	jp.terasoluna.fw.batch.restart.JobRestartTableHandler	ジョブリスタート管理テーブルのハンドラクラス用Bean定義。	queryDAO updateDAO jobRestartInfoFactory	データベースBean定義ファイルを参照 データベースBean定義ファイルを参照 項番22を参照
21	ジョブリスタート情報用パラメータクラス	JobRestartInfoFactory	jp.terasoluna.fw.batch.restart.StandardJobRestartInfoFactory	ジョブリスタート情報用パラメータクラスのBean定義。	なし	なし

チャンク別トランザクションモデル (ブレイク)

順番	設定項目名称	Bean ID	Classes/parent名	概要	property	property概要
0	ブレースホルダ			ブレースホルダの定義	import resource	初期値 [classpath:common/PlaceholderConfig.xml]
1	マネージャ	jobManager	baseManager (jp.terasoluna.fw.batch.core.JobManager)	ジョブ実行マネージャの定義。	name collector workQueueFactory preProcessor postProcessor	初期値 [chunkTransactionJobManager] ジョブBean定義ファイルを参照 項番2を参照 項番3を参照 項番5を参照
2		-	jp.terasoluna.fw.batch.common.WorkManagerTaskWorkQueueFactory	common対応作業キュー生成クラスの定義。	workerExecuterService	同期型起動用スレッドプール定義ファイルを参照 非同期型起動用スレッドプール定義ファイルを参照
3		jobPreProcessor	transactionalSupportProcessor (jp.terasoluna.fw.batch.spring.support.transactionalSupportProcessor)	トランザクション制御有りのジョブ前処理の定義。	queueLength queueProcessor supportProcessor	デフォルトBean定義ファイルを参照 項番7を参照 項番4を参照
4		-	standardSupportProcessor (jp.terasoluna.fw.batch.standard.StandardSupportProcessor)	ジョブ前処理の定義。	supportLogicList	デフォルトBean定義ファイルを参照 ジョブBean定義ファイルを参照
5		jobPostProcessor	transactionalSupportProcessor (jp.terasoluna.fw.batch.spring.support.transactionalSupportProcessor)	トランザクション制御有りのジョブ後処理の定義。	supportProcessor	項番6を参照
6		-	standardSupportProcessor (jp.terasoluna.fw.batch.standard.StandardSupportProcessor)	ジョブ後処理の定義。	supportLogicList	デフォルトBean定義ファイルを参照 ジョブBean定義ファイルを参照
7	キュープロセッサ	chunkTransactionQueueProcessor	baseQueueProcessor (jp.terasoluna.fw.batch.standard.QueueProcessor)	キュー処理クラスの定義。	name worker	初期値 [chunkTransactionQueueProcessor] 項番8を参照
8	ワーカ	chunkTransactionWorker	TransactionWorker	トランザクション制御有りのワーカ定義。	transactionManager useSavepoint jobWorker	データベースBean定義ファイルを参照 デフォルトBean定義ファイルを参照 ジョブBean定義ファイルを参照 項番9を参照
9		controlBreakWorker	jp.terasoluna.fw.batch.core.JobWorker	チャンク処理クラスの定義。	logicExecutor batchUpdateProcessor exceptionHandlerMap batchUpdateResultHandler defaultJobExceptionHandler	項番10を参照 項番13を参照 デフォルトBean定義ファイルを参照 フレームワークBean定義ファイルを参照 デフォルトBean定義ファイルを参照
10		controlBreakLogicExecutor	jp.terasoluna.fw.batch.controlbreak.ControlBreakLogicExecutor	コントロールブレイク用ビジネスロジック実行クラスの定義。	controlBreakProcessor	項番12を参照 項番11を参照
11		controlBreakProcessor	jp.terasoluna.fw.batch.controlbreak.ControlBreakProcessor	コントロールブレイク処理実行クラスの定義。	controlBreakDef logicResultHandler exceptionHandlerMap defaultJobExceptionHandler throwableHandler	項番14を参照 デフォルトBean定義ファイルを参照 デフォルトBean定義ファイルを参照 デフォルトBean定義ファイルを参照 ThrowableハンドラのBean定義。
12		logicExecutor	jp.terasoluna.fw.batch.core.StandardLogicExecutor	ビジネスロジック実行クラスの定義。	throwableHandler	初期設定値は [throwableHandler] である。 デフォルトBean定義ファイルを参照。
13		batchUpdateProcessor	jp.terasoluna.fw.batch.standard.StandardBatchUpdateProcessor	バッチ更新処理実行クラスの定義。	updateDAO	データベースBean定義ファイルを参照
14		controlBreakDef	jp.terasoluna.fw.batch.controlbreak.ControlBreakDef	コントロールブレイク情報クラスの定義。	controlBreakDefItemList chunkControlBreakDefItem transChunkControlBreakDefItemList	ジョブBean定義ファイルを参照 ジョブBean定義ファイルを参照 ジョブBean定義ファイルを参照
12	ジョブの処理状況クラス	MonitorableJobStatus	jp.terasoluna.fw.batch.monitor.MonitorableJobStatus	監視用ジョブの処理状況用Bean定義。	beanRegister transactionManager useSavepoint partitionNo defaultJobExitCodeMap	フレームワークBean定義ファイルを参照 データベースBean定義ファイルを参照 デフォルトBean定義ファイルを参照 初期値 [1] デフォルトBean定義ファイルを参照
13		JobStatus	jp.terasoluna.fw.batch.core.JobStatus	ジョブの処理状況用Bean定義。	partitionNo transactionManager useSavepoint partitionNo defaultJobExitCodeMap	初期値 [1] データベースBean定義ファイルを参照 デフォルトBean定義ファイルを参照 初期値 [1] デフォルトBean定義ファイルを参照
14	スレッドプールサイズ	threadSize	java.lang.Integer	スレッドプールのサイズ	constructor-arg	初期値 [1]

非トランザクションモデル

項番	設定項目名称	Bean ID	Classes/parent名	概要	property	property概要
0	ブレースホルダ			ブレースホルダの定義	import resource	初期値 [classpath:common/PlaceholderConfig.xml]
1	マネージャ定義	jobManager	baseManager (jp.terasoluna.fw.batch.core.JobManager)	ジョブ実行マネージャの定義。	name	初期値 [noTransactionJobManager]
					collector	ジョブBean定義ファイルを参照
					workQueueFactory	項番2を参照
					preProcessor	項番3を参照
2		-	jp.terasoluna.fw.batch.common.WorkManagerTaskWorkerQueueFactory	commonに対応作業キュー生成クラスの定義。	postProcessor	項番4を参照
					workerExecutorService	同期型起動用スレッドプール定義ファイルを参照
					queueLength	非同期型起動用スレッドプール定義ファイルを参照
3		jobPreProcessor	standardSupportProcessor (jp.terasoluna.fw.batch.standard.StandardSupportProcessor)	ジョブ前処理の定義。	queueProcessor	デフォルトBean定義ファイルを参照
4		jobPostProcessor	standardSupportProcessor (jp.terasoluna.fw.batch.standard.StandardSupportProcessor)	ジョブ後処理の定義。	supportLogicList	デフォルトBean定義ファイルを参照
5	キュープロセッサ	noTransactionQueueProcessor	baseQueueProcessor (jp.terasoluna.fw.batch.standard.QueueProcessor)	キュー処理クラスの定義。	supportLogicList	ジョブBean定義ファイルを参照
6	ワーカ	noTransactionWorker	baseWorker (jp.terasoluna.fw.batch.core.JobWorker)	チャンク処理クラスの定義。	name	初期値 [noTransactionQueueProcessor]
					worker	項番6を参照
					biLogicExecutor	項番7を参照
					exceptionHandlerMap	デフォルトBean定義ファイルを参照
7		biLogicExecutor	jp.terasoluna.fw.batch.core.StandardBiLogicExecutor	ビジネスロジック実行クラスの定義。	defaultJobExceptionHandler	デフォルトBean定義ファイルを参照
					batchUpdateProcessor	項番8を参照
					batchUpdateResultHandler	デフォルトBean定義ファイルを参照
8		batchUpdateProcessor	jp.terasoluna.fw.batch.standard.StandardBatchUpdateProcessor	バッチ更新処理実行クラスの定義。	biLogic	ジョブBean定義ファイルを参照
					biLogicResultHandler	デフォルトBean定義ファイルを参照
					exceptionHandlerMap	デフォルトBean定義ファイルを参照
					defaultJobExceptionHandler	デフォルトBean定義ファイルを参照
					throwableHandler	ThrowableハンドラのBean定義。 初期設定値は [throwableHandler] である。
9	ジョブの処理状況クラス	MonitorableJobStatus	jp.terasoluna.fw.batch.monitor.MonitorableJobStatus	監視用ジョブの処理状況用Bean定義。	updateDAO	デフォルトBean定義ファイルを参照
					mbeanRegister	データアクセスBean定義ファイルを参照
					transactionManager	データアクセスBean定義ファイルを参照
					useSavepoint	デフォルトBean定義ファイルを参照
					partitionNo	初期値 [1]
					defaultJobExitCodeMap	デフォルトBean定義ファイルを参照
10		JobStatus	jp.terasoluna.fw.batch.core.JobStatus	ジョブの処理状況用Bean定義。	partitionNo	初期値 [1]
					transactionManager	データアクセスBean定義ファイルを参照
					useSavepoint	デフォルトBean定義ファイルを参照
					partitionNo	初期値 [1]
					defaultJobExitCodeMap	デフォルトBean定義ファイルを参照
11	スレッドプールサイズ	threadSize	java.lang.Integer	スレッドプールのサイズ	constructor-arg	初期値 [1]

単ートランザクションモデル

順番	設定項目名称	Bean ID	Classes/parent名	概要	property	property概要
0	ブレースホルダ			ブレースホルダの定義	import resource	初期値 [classpath:common/PlaceholderConfig.xml]
1	マネージャ	jobManager	baseManager (jp.terasoluna.fw.batch.core.JobManager)	ジョブ実行マネージャの定義。	name collector workQueueFactory preProcessor postProcessor	初期値 [singleTransactionJobManager] ジョブBean定義ファイルを参照 項番2を参照 項番3を参照 項番5を参照
2		-	jp.terasoluna.fw.batch.common.WorkManagerTaskWorkQueueFactory	common)対応作業キュー生成クラスの定義。	workerExecutorService queueLength queueProcessor	同期型起動用スレッドプール定義ファイルを参照 非同期型起動用スレッドプール定義ファイルを参照 デフォルトBean定義ファイルを参照 項番7を参照
3		jobPreProcessor	transactionalSupportProcessor (jp.terasoluna.fw.batch.spring.support.transactionalSupportProcessor)	トランザクション制御有りのジョブ前処理の定義。	supportProcessor	項番4を参照
4		-	standardSupportProcessor (jp.terasoluna.fw.batch.standard.StandardSupportProcessor)	ジョブ前処理の定義。	supportLogicList	デフォルトBean定義ファイルを参照 ジョブBean定義ファイルを参照
5		jobPostProcessor	transactionalSupportProcessor (jp.terasoluna.fw.batch.spring.support.transactionalSupportProcessor)	トランザクション制御有りのジョブ後処理の定義。	supportProcessor	項番6を参照
6		-	standardSupportProcessor (jp.terasoluna.fw.batch.standard.StandardSupportProcessor)	ジョブ後処理の定義。	supportLogicList	デフォルトBean定義ファイルを参照 ジョブBean定義ファイルを参照
7	キュープロセッサ	singleTransactionQueueProcessor	jp.terasoluna.fw.batch.spring.support.transactionalSupportQueueProcessor	キュー処理クラスの定義。	name worker transactionManager preProcessor postProcessor throwableHandler	初期値 [singleTransactionQueueProcessor] 項番10を参照 データアクセスBean定義ファイルを参照 項番8を参照 項番9を参照 ThrowableハンドラのBean定義。 初期設定値は [throwableHandler] である。 デフォルトBean定義ファイルを参照
8		firstChunkPreProcessor	standardSupportProcessor (jp.terasoluna.fw.batch.standard.StandardSupportProcessor)	先頭チャンク前処理の定義。	supportLogicList	デフォルトBean定義ファイルを参照 ジョブBean定義ファイルを参照
9		lastChunkPostProcessor	standardSupportProcessor (jp.terasoluna.fw.batch.standard.StandardSupportProcessor)	最終チャンク後処理の定義。	supportLogicList	デフォルトBean定義ファイルを参照 ジョブBean定義ファイルを参照
10	ワーカ	singleTransactionWorker	jp.terasoluna.fw.batch.core.JobWorker	チャンク処理クラスの定義。	blologicExecutor batchUpdateProcessor exceptionHandlerMap batchUpdateResultHandler defaultJobExceptionHandler throwableHandler	項番11を参照 項番12を参照 デフォルトBean定義ファイルを参照 デフォルトBean定義ファイルを参照 デフォルトBean定義ファイルを参照 ThrowableハンドラのBean定義。 初期設定値は [throwableHandler] である。 デフォルトBean定義ファイルを参照。
11		blologicExecutor	jp.terasoluna.fw.batch.core.StandardBlologicExecutor	ビジネスロジック実行クラスの定義。	blologic blologicResultHandler exceptionHandlerMap defaultJobExceptionHandler throwableHandler	ジョブBean定義ファイルを参照 デフォルトBean定義ファイルを参照 デフォルトBean定義ファイルを参照 デフォルトBean定義ファイルを参照 ThrowableハンドラのBean定義。 初期設定値は [throwableHandler] である。 デフォルトBean定義ファイルを参照。
12		batchUpdateProcessor	jp.terasoluna.fw.batch.standard.StandardBatchUpdateProcessor	バッチ更新処理クラスの定義。	updateDAO	データアクセスBean定義ファイルを参照
13	ジョブの処理状況クラス	MonitorableJobStatus	jp.terasoluna.fw.batch.monitor.MonitorableJobStatus	監視用ジョブの処理状況用Bean定義。	mbeanRegister transactionManager useSavepoint partitionNo defaultJobExitCodeMap	フレームワークBean定義ファイルを参照 データアクセスBean定義ファイルを参照 デフォルトBean定義ファイルを参照 初期値 [1] デフォルトBean定義ファイルを参照
14		JobStatus	jp.terasoluna.fw.batch.core.JobStatus	ジョブの処理状況用Bean定義。	partitionNo transactionManager useSavepoint partitionNo defaultJobExitCodeMap	初期値 [1] データアクセスBean定義ファイルを参照 デフォルトBean定義ファイルを参照 初期値 [1] デフォルトBean定義ファイルを参照
15	スレッドプールサイズ	threadSize	java.lang.Integer	スレッドプールのサイズ	constructor-arg	初期値 [1]

ジョブ分割チャンク別トランザクションモデル

項番	設定項目名称	Bean ID	Classes/parent名	概要	property	property概要
0	ブレースホルダ			ブレースホルダの定義	import resource	初期値 [classpath:common/PlaceholderConfig.xml]
1	親ジョブマネージャ	jobManager	baseManager (jp.terasoluna.fw.batch.core.JobManager)	親ジョブ実行用マネージャ定義。	name collector workQueueFactory preProcessor postProcessor	初期値 [partitionChunkTransactionJobManager] ジョブBean定義ファイルを参照 項番2を参照 項番3を参照 項番5を参照
2		-	jp.terasoluna.fw.batch.common.WorkManagerTaskWorkerQueueFactory	commonに対応親ジョブ用作業キュー作成クラスの定義。	workerExecutorService	同期型起動用スレッドプール定義ファイルを参照 非同期型起動用スレッドプール定義ファイルを参照
3		parentJobPreProcessor	transactionalSupportProcessor (jp.terasoluna.fw.batch.springsupport.transactionalSupportProcessor)	トランザクション制御有りの親ジョブ前処理の定義。	queueLength queueProcessor multiplicity	デフォルトBean定義ファイルを参照 項番7を参照 ジョブBean定義ファイルを参照 項番4を参照
4		-	standardSupportProcessor (jp.terasoluna.fw.batch.standard.StandardSupportProcessor)	親ジョブ前処理の定義。	supportLogicList	デフォルトBean定義ファイルを参照 ジョブBean定義ファイルを参照
5		parentJobPostProcessor	transactionalSupportProcessor (jp.terasoluna.fw.batch.springsupport.transactionalSupportProcessor)	トランザクション制御有りの親ジョブ後処理の定義。	supportProcessor	項番6を参照
6		-	standardSupportProcessor (jp.terasoluna.fw.batch.standard.StandardSupportProcessor)	親ジョブ後処理の定義。	supportLogicList	デフォルトBean定義ファイルを参照 ジョブBean定義ファイルを参照
7	親ジョブキュープロセッサ	partitionChunkTransactionQueueProcessor	baseQueueProcessor (jp.terasoluna.fw.batch.standard.QueueProcessor)	親ジョブ用キュー処理クラスの定義。	name	初期値 [partitionChunkTransactionQueueProcessor]
8	子ジョブマネージャ	childJobManager	baseManager (jp.terasoluna.fw.batch.core.JobManager)	子ジョブ実行用マネージャ定義。	worker name collector workQueueFactory preProcessor postProcessor	項番8を参照 初期値 [childJobManager] ジョブBean定義ファイルを参照 項番9を参照 項番10を参照 項番12を参照
9		-	jp.terasoluna.fw.batch.common.WorkManagerTaskWorkerQueueFactory	commonに対応子ジョブ用作業キュー作成クラスの定義。	workerExecutorService	同期型起動用スレッドプール定義ファイルを参照 非同期型起動用スレッドプール定義ファイルを参照
10		jobPreProcessor	transactionalSupportProcessor (jp.terasoluna.fw.batch.springsupport.transactionalSupportProcessor)	トランザクション制御有りの子ジョブ前処理の定義。	queueLength queueProcessor supportProcessor	デフォルトBean定義ファイルを参照 項番14を参照 項番11を参照
11		-	standardSupportProcessor (jp.terasoluna.fw.batch.standard.StandardSupportProcessor)	子ジョブ前処理の定義。	supportLogicList	デフォルトBean定義ファイルを参照 ジョブBean定義ファイルを参照
12		jobPostProcessor	transactionalSupportProcessor (jp.terasoluna.fw.batch.springsupport.transactionalSupportProcessor)	トランザクション制御有りの子ジョブ後処理の定義。	supportProcessor	項番13を参照
13		-	standardSupportProcessor (jp.terasoluna.fw.batch.standard.StandardSupportProcessor)	子ジョブ後処理の定義。	supportLogicList	デフォルトBean定義ファイルを参照 ジョブBean定義ファイルを参照
14	子ジョブキュープロセッサ	partitionChunkTransactionChildQueueProcessor	baseQueueProcessor (jp.terasoluna.fw.batch.standard.QueueProcessor)	子ジョブ用キュー処理クラスの定義。	name	初期値 [partitionChunkTransactionChildQueueProcessor]
15	ワーカ	transactionWorker	jp.terasoluna.fw.batch.springsupport.transactionalTransactionWorker	トランザクション制御有りのワーカ定義。	worker transactionManager useSavepoint jobWorker	項番15を参照 データアクセスBean定義ファイルを参照 デフォルトBean定義ファイルを参照 ジョブBean定義ファイルを参照 項番16を参照
16		chunkWorker	jp.terasoluna.fw.batch.core.JobWorker	チャンク処理クラスの定義。	logicExecutor batchUpdateProcessor exceptionHandlerMap	項番18を参照 項番17を参照 デフォルトBean定義ファイルを参照
17		batchUpdateProcessor	jp.terasoluna.fw.batch.standard.StandardBatchUpdateProcessor	バッチ更新処理クラスの定義。	batchUpdateResultHandler defaultJobExceptionHandler throwableHandler	デフォルトBean定義ファイルを参照 デフォルトBean定義ファイルを参照 ThrowableハンドラのBean定義。 初期設定値は [throwableHandler] である。 デフォルトBean定義ファイルを参照
18		batchLogicExecutor	jp.terasoluna.fw.batch.core.StandardBatchLogicExecutor	ビジネスロジック実行クラスの定義。	updateDAO	データアクセスBean定義ファイルを参照
19		monitorableJobStatus	jp.terasoluna.fw.batch.monitor.MonitorableJobStatus	監視用ジョブの処理状況用Bean定義。	logic logicResultHandler exceptionHandlerMap defaultJobExceptionHandler throwableHandler	ジョブBean定義ファイルを参照 デフォルトBean定義ファイルを参照 デフォルトBean定義ファイルを参照 デフォルトBean定義ファイルを参照 ThrowableハンドラのBean定義。 初期設定値は [throwableHandler] である。 デフォルトBean定義ファイルを参照
20		jobStatus	jp.terasoluna.fw.batch.core.JobStatus	ジョブの処理状況用Bean定義。	throwableHandler	フレームワークBean定義ファイルを参照 データアクセスBean定義ファイルを参照 デフォルトBean定義ファイルを参照 デフォルトBean定義ファイルを参照 初期値 [i] デフォルトBean定義ファイルを参照
21	スレッドプールサイズ	threadSize	java.lang.Integer	スレッドプールのサイズ	partitionNo transactionManager useSavepoint partitionNo defaultJobExitCodeMap multiplicity	初期値 [i] データアクセスBean定義ファイルを参照 デフォルトBean定義ファイルを参照 初期値 [i] デフォルトBean定義ファイルを参照 デフォルトBean定義ファイルを参照

ジョブ分割チャック別トランザクションモデル (リスタート)

項番	設定項目名称	Bean ID	Classes/parent名	概要	property	property概要
0	ブレースホルダ	-	-	ブレースホルダの定義	import_resource	初期値 [classpath:common/PlaceholderConfig.xml]
1	jobManager	jp.terasoluna.fw.batch.restart.JobContextRestorer	-	親ジョブリスタート実行用マネージャの定義。	jobRestartTableHandler throwableHandler	項番31を参照 項番2を参照 ThrowableハンドラのBean定義。 初期設定値は [throwableHandler] である。 デフォルトBean定義ファイルを参照。
2	delegatedJobManager	baseManager (jp.terasoluna.fw.batch.core.JobManager)	-	親ジョブ実行用マネージャ定義。	name collector workQueueFactory preProcessor postProcessor	初期値 [partitionChunkTransactionJobManager] ジョブBean定義ファイルを参照 項番3を参照 項番4を参照 項番1を参照
3	-	jp.terasoluna.fw.batch.commonj.WorkManagerTaskWorkQueueFactory	commonj対応親ジョブ用作業キュー作成クラスの定義。	workerExecutorService	同期型起動用スレッドプール定義ファイルを参照 非同期型起動用スレッドプール定義ファイルを参照	
4	parentJobPreProcessor	jp.terasoluna.fw.batch.restart.RestartPreProcessor	リスタート時の親ジョブ前処理の定義。	queueLength multiplicity queueProcessor jobRestartTableHandler	デフォルトBean定義ファイルを参照 ジョブBean定義ファイルを参照 項番10を参照 項番31を参照	
5	親ジョブマネージャ	transactionalSupportProcessor (jp.terasoluna.fw.batch.spring.support.transaction.TransactionSupportProcessor) standardSupportProcessor (jp.terasoluna.fw.batch.standard.StandardSupportProcessor)	トランザクション管理有りの親ジョブ前処理の定義。	preProcessor parentManager throwableHandler	項番5を参照 初期値 [true] ThrowableハンドラのBean定義。 初期設定値は [throwableHandler] である。 デフォルトBean定義ファイルを参照。	
6	-	jp.terasoluna.fw.batch.standard.StandardSupportProcessor	親ジョブ前処理の定義。	supportLogicList	デフォルトBean定義ファイルを参照 ジョブBean定義ファイルを参照	
7	parentJobPostProcessor	jp.terasoluna.fw.batch.restart.RestartPostProcessor baseQueueProcessor (jp.terasoluna.fw.batch.standard.QueueProcessor)	リスタート時の親ジョブ後処理の定義。	jobRestartTableHandler postProcessor sqlKey parentManager throwableHandler	項番31を参照 項番8を参照 リスタート情報クリア用SQL 初期設定値は [jobRestart.DELETE_JOB_RESTART_CLEAR] である。 初期値 [true] ThrowableハンドラのBean定義。 初期設定値は [throwableHandler] である。 デフォルトBean定義ファイルを参照。	
8	delegatedParentJobPostProcessor	transactionalSupportProcessor (jp.terasoluna.fw.batch.spring.support.transaction.TransactionSupportProcessor) standardSupportProcessor (jp.terasoluna.fw.batch.standard.StandardSupportProcessor)	トランザクション管理有りの親ジョブ後処理の定義。	supportProcessor	項番9を参照	
9	-	jp.terasoluna.fw.batch.standard.StandardSupportProcessor	親ジョブ後処理の定義。	supportLogicList	デフォルトBean定義ファイルを参照 ジョブBean定義ファイルを参照	
10	親ジョブキューロセッサ	partitionChunkTransactionQueueProcessor (jp.terasoluna.fw.batch.standard.QueueProcessor)	親ジョブ用キュー処理クラスの定義。	name worker	初期値 [partitionChunkTransactionQueueProcessor] 項番11を参照	
11	restartChildJobManager	jp.terasoluna.fw.batch.restart.JobContextRestorer	子ジョブリスタート実行用マネージャの定義。	jobRestartTableHandler jobManager throwableHandler	項番31を参照 項番12を参照 ThrowableハンドラのBean定義。 初期設定値は [throwableHandler] である。 デフォルトBean定義ファイルを参照。	
12	restratDelegatedChildJobManager	baseManager (jp.terasoluna.fw.batch.core.JobManager)	子ジョブ実行用マネージャ定義。	name collector workQueueFactory preProcessor postProcessor workerExecutorService	初期値 [childJobManager] ジョブBean定義ファイルを参照 項番13を参照 項番14を参照 項番17を参照 同期型起動用スレッドプール定義ファイルを参照 非同期型起動用スレッドプール定義ファイルを参照	
13	-	jp.terasoluna.fw.batch.commonj.WorkManagerTaskWorkQueueFactory	commonj対応子ジョブ用作業キュー作成クラスの定義。	queueLength queueProcessor jobRestartTableHandler	デフォルトBean定義ファイルを参照 項番20を参照 項番31を参照	
14	jobPreProcessor	jp.terasoluna.fw.batch.restart.RestartPreProcessor	リスタート時の子ジョブ前処理の定義。	preProcessor throwableHandler	項番15を参照 ThrowableハンドラのBean定義。 初期設定値は [throwableHandler] である。 デフォルトBean定義ファイルを参照。	
15	子ジョブマネージャ	transactionalSupportProcessor (jp.terasoluna.fw.batch.spring.support.transaction.TransactionSupportProcessor) standardSupportProcessor (jp.terasoluna.fw.batch.standard.StandardSupportProcessor)	トランザクション管理有りの子ジョブ前処理の定義。	supportProcessor	項番16を参照	
16	-	jp.terasoluna.fw.batch.standard.StandardSupportProcessor	子ジョブ前処理の定義。	supportLogicList	デフォルトBean定義ファイルを参照 ジョブBean定義ファイルを参照	
17	jobPostProcessor	jp.terasoluna.fw.batch.restart.RestartPostProcessor baseQueueProcessor (jp.terasoluna.fw.batch.standard.QueueProcessor)	リスタート時の子ジョブ後処理の定義。	jobRestartTableHandler postProcessor sqlKey throwableHandler	デフォルトBean定義ファイルを参照 項番18を参照 リスタート情報クリア用SQL 初期設定値は [jobRestart.UPDATE_JOB_RESTART_CLEAR] である。 ThrowableハンドラのBean定義。 初期設定値は [throwableHandler] である。 デフォルトBean定義ファイルを参照。	
18	delegatedJobPostProcessor	transactionalSupportProcessor (jp.terasoluna.fw.batch.spring.support.transaction.TransactionSupportProcessor) standardSupportProcessor (jp.terasoluna.fw.batch.standard.StandardSupportProcessor)	トランザクション管理有りの子ジョブ後処理の定義。	supportProcessor	項番19を参照	
19	-	jp.terasoluna.fw.batch.standard.StandardSupportProcessor	子ジョブ後処理の定義。	supportLogicList	デフォルトBean定義ファイルを参照 ジョブBean定義ファイルを参照	
20	子ジョブキューロセッサ	partitionChunkTransactionChildQueueProcessor (jp.terasoluna.fw.batch.standard.QueueProcessor)	子ジョブ用キュー処理クラスの定義。	name worker	初期値 [partitionChunkTransactionChildQueueProcessor] 項番21を参照	
21	transactionWorker	jp.terasoluna.fw.batch.spring.support.transaction.TransactionWorker	チャック処理クラスの定義。	transactionManager useSavepoint	データベースBean定義ファイルを参照 デフォルトBean定義ファイルを参照 ジョブBean定義ファイルを参照	
22	restartPointUpdater	jp.terasoluna.fw.batch.restart.RestartPointUpdater	リスタートポイント登録用ワーククラスの定義。	jobWorker jobRestartTableHandler throwableHandler	項番22を参照 項番31を参照 ThrowableハンドラのBean定義。 初期設定値は [throwableHandler] である。 デフォルトBean定義ファイルを参照。	
23	restartPointCloser	jp.terasoluna.fw.batch.restart.RestartPointCloser	リスタートポイント削除用ワーククラスの定義。	jobWorker jobRestartTableHandler supportLogicList sqlKey throwableHandler	項番24を参照 項番31を参照 デフォルトBean定義ファイルを参照 ジョブBean定義ファイルを参照 リスタート情報クリア用SQL 初期設定値は [jobRestart.UPDATE_JOB_RESTART_CLEAR] である。 ThrowableハンドラのBean定義。 初期設定値は [throwableHandler] である。 デフォルトBean定義ファイルを参照。	
24	chunkWorker	jp.terasoluna.fw.batch.core.JobWorker	チャック処理クラスの定義。	blockExecutor batchUpdateProcessor exceptionHandlerMap batchJobExceptionHandler defaultJobExceptionHandler throwableHandler	項番25を参照 項番26を参照 デフォルトBean定義ファイルを参照 デフォルトBean定義ファイルを参照 デフォルトBean定義ファイルを参照 ThrowableハンドラのBean定義。 初期設定値は [throwableHandler] である。 デフォルトBean定義ファイルを参照。	
25	batchUpdateProcessor	jp.terasoluna.fw.batch.standard.StandardBatchUpdateProcessor	バッチ更新処理実行クラスの定義。	updateDAO	データベースBean定義ファイルを参照	
26	blockExecutor	jp.terasoluna.fw.batch.core.StandardBlockExecutor	ビジネスロジック実行クラスの定義。	block blockResultHandler exceptionHandlerMap defaultJobExceptionHandler throwableHandler	ジョブBean定義ファイルを参照 デフォルトBean定義ファイルを参照 デフォルトBean定義ファイルを参照 デフォルトBean定義ファイルを参照 ThrowableハンドラのBean定義。 初期設定値は [throwableHandler] である。 デフォルトBean定義ファイルを参照。	
27	ジョブの処理状況クラス	MonitorableJobStatus (jp.terasoluna.fw.batch.monitor.MonitorableJobStatus)	監視用ジョブの処理状況用Bean定義。	mbeanRegister transactionManager useSavepoint partitionNo defaultJobExitCodeMap	フレームワークBean定義ファイルを参照 データベースBean定義ファイルを参照 デフォルトBean定義ファイルを参照 初期値 [-1] デフォルトBean定義ファイルを参照	
28	JobStatus	jp.terasoluna.fw.batch.core.JobStatus	ジョブの処理状況用Bean定義。	partitionNo transactionManager useSavepoint partitionNo defaultJobExitCodeMap multiplicity	初期値 [-1] データベースBean定義ファイルを参照 デフォルトBean定義ファイルを参照 初期値 [-1] デフォルトBean定義ファイルを参照 ジョブBean定義ファイルを参照	
29	スレッドプールサイズ	threadSize	スレッドプールのサイズ	queryDAO updateDAO	データベースBean定義ファイルを参照 データベースBean定義ファイルを参照	
30	ジョブリスタート管理テーブルハンドラ	JobRestartTableHandler	ジョブリスタート管理テーブルのハンドラクラス用Bean定義。	jobRestartInfoFactory	項番22を参照 なし	
31	ジョブリスタート情報用パラメータクラス	JobRestartInfoFactory	ジョブリスタート情報用パラメータクラスのBean定義。	なし	なし	

ジョブ分割非トランザクションモデル

順番	設定項目名称	Bean ID	Classes/parent名	概要	property	property概要
0	ブレースホルダ			ブレースホルダの定義	import resource	初期値 [classpath:common/PlaceholderConfig.xml]
1		jobManager	baseManager (jp.terasoluna.fw.batch.core.JobManager)	親ジョブ実行用マネージャの定義。	name	初期値 [partitionNoTransactionJobManager]
2	親ジョブマネージャ	-	jp.terasoluna.fw.batch.common.WorkManagerTaskWorkerQueueFactory	common対応親ジョブ用作業キュー作成クラスの定義。	collector	ジョブBean定義ファイルを参照
3					workQueueFactory	項目2を参照
4					preProcessor	項目3を参照
5					postProcessor	項目4を参照
6					workerExecutorService	同期型起動用スレッドプール定義ファイルを参照 非同期型起動用スレッドプール定義ファイルを参照
7					queueProcessor	項目5を参照
8					queueLength	デフォルトBean定義ファイルを参照
9					multiplicity	ジョブBean定義ファイルを参照
10					supportLogicList	デフォルトBean定義ファイルを参照 ジョブBean定義ファイルを参照
11					supportLogicList	デフォルトBean定義ファイルを参照 ジョブBean定義ファイルを参照
12					name	初期値 [partitionNoTransactionQueueProcessor]
13					worker	項目6を参照
14					name	初期値 [childJobManager]
15					collector	ジョブBean定義ファイルを参照
16					workQueueFactory	項目7を参照
17					preProcessor	項目8を参照
18					postProcessor	項目9を参照
19					workerExecutorService	同期型起動用スレッドプール定義ファイルを参照 非同期型起動用スレッドプール定義ファイルを参照
20					queueLength	デフォルトBean定義ファイルを参照
21					queueProcessor	項目10を参照
22					supportLogicList	デフォルトBean定義ファイルを参照 ジョブBean定義ファイルを参照
23					supportLogicList	デフォルトBean定義ファイルを参照 ジョブBean定義ファイルを参照
24					name	初期値 [partitionNoTransactionChildQueueProcessor]
25					worker	項目11を参照
26					biogicExecutor	項目12を参照
27					exceptionHandlerMap	デフォルトBean定義ファイルを参照
28					defaultJobExceptionHandler	デフォルトBean定義ファイルを参照
29					batchUpdateProcessor	項目13を参照
30					batchUpdateResultHandler	フレームワークBean定義ファイルを参照
31					throwableHandler	ThrowableハンドラのBean定義。 初期設定値は [throwableHandler] である。 デフォルトBean定義ファイルを参照。
32					biogicResultHandler	ジョブBean定義ファイルを参照
33					exceptionHandlerMap	デフォルトBean定義ファイルを参照
34					defaultJobExceptionHandler	デフォルトBean定義ファイルを参照
35					throwableHandler	ThrowableハンドラのBean定義。 初期設定値は [throwableHandler] である。 デフォルトBean定義ファイルを参照。
36					updateDAO	データアクセスBeanファイルを参照
37					mbeanRegister	フレームワークBean定義ファイルを参照
38					partitionNo	初期値 [1]
39					defaultJobExitCodeMap	デフォルトBean定義ファイルを参照
40					partitionNo	初期値 [1]
41					partitionNo	初期値 [1]
42					defaultJobExitCodeMap	デフォルトBean定義ファイルを参照
43					multiplicity	ジョブBean定義ファイルを参照



ジョブ分割単一トランザクションモデル

順番	設定項目名称	Bean ID	Classes/parent名	概要	property	property概要
0	ブレースホルダ			ブレースホルダの定義	import resource	初期値『classpath:common/PlaceholderConfig.xml』
1	親ジョブマネージャ	jobManager	baseManager (jp.terasoluna.fw.batch.core.JobManager)	親ジョブ実行用マネージャの定義。	name	初期値 『partitionSingleTransactionJobManager』
2		—	jp.terasoluna.fw.batch.common.JobManagerTaskWorkerQueueFactory	common)対応親ジョブ用作業キュー作成クラスの定義。	collector	ジョブBean定義ファイルを参照
3		parentJobPreProcessor	transactionalSupportProcessor (jp.terasoluna.fw.batch.spring.support.transactionalSupportProcessor)	トランザクション管理有りの親ジョブ前処理の定義。	workQueueFactory	項番2を参照
4		—	standardSupportProcessor (jp.terasoluna.fw.batch.standard.StandardSupportProcessor)	親ジョブ前処理の定義。	workProcessor	項番3を参照
5		parentJobPostProcessor	transactionalSupportProcessor (jp.terasoluna.fw.batch.spring.support.transactionalSupportProcessor)	トランザクション管理有りの親ジョブ後処理の定義。	postProcessor	項番5を参照
6		—	standardSupportProcessor (jp.terasoluna.fw.batch.standard.StandardSupportProcessor)	親ジョブ後処理の定義。	workerExecutorService	同期型起動用スレッドプール定義ファイルを参照 非同同期型起動用スレッドプール定義ファイルを参照
7	親ジョブキュープロセッサ	partitionSingleTransactionQueueProcessor	baseQueueProcessor (jp.terasoluna.fw.batch.standard.QueueProcessor)	親ジョブ用キュー処理クラスの定義。	queueProcessor	項番7を参照
8	子ジョブマネージャ	childJobManager	baseManager (jp.terasoluna.fw.batch.core.JobManager)	子ジョブ実行用マネージャの定義。	queueLength	デフォルトBean定義ファイルを参照 ジョブBean定義ファイルを参照
9		—	jp.terasoluna.fw.batch.common.JobManagerTaskWorkerQueueFactory	common)対応子ジョブ用作業キュー作成クラスの定義。	multiplicity	ジョブBean定義ファイルを参照
10		jobPreProcessor	transactionalSupportProcessor (jp.terasoluna.fw.batch.spring.support.transactionalSupportProcessor)	トランザクション管理有りの子ジョブ前処理の定義。	supportProcessor	項番4を参照
11		—	standardSupportProcessor (jp.terasoluna.fw.batch.standard.StandardSupportProcessor)	子ジョブ前処理の定義。		
12		jobPostProcessor	transactionalSupportProcessor (jp.terasoluna.fw.batch.spring.support.transactionalSupportProcessor)	トランザクション管理有りの子ジョブ後処理の定義。		
13		—	standardSupportProcessor (jp.terasoluna.fw.batch.standard.StandardSupportProcessor)	子ジョブ後処理の定義。		
14	子ジョブキュープロセッサ	singleTransactionChildQueueProcessor	baseQueueProcessor (jp.terasoluna.fw.batch.standard.QueueProcessor)	子ジョブ用キュー処理クラスの定義。	name	初期値 『partitionSingleTransactionChildQueueProcessor』
15		firstChunkPreProcessor	standardSupportProcessor (jp.terasoluna.fw.batch.standard.StandardSupportProcessor)	先頭チャンク前処理の定義。	worker	項番8を参照 初期値『childJobManager』
16		lastChunkPostProcessor	standardSupportProcessor (jp.terasoluna.fw.batch.standard.StandardSupportProcessor)	最終チャンク後処理の定義。	collector	ジョブBean定義ファイルを参照
17	ワーカ	singleTransactionWorker	jp.terasoluna.fw.batch.core.JobWorker	チャンク処理クラスの定義。	workQueueFactory	項番9を参照
18		batchUpdateProcessor	jp.terasoluna.fw.batch.standard.StandardBatchUpdateProcessor	バッチ更新処理クラスの定義。	preProcessor	項番10を参照
19		batchLogicExecutor	jp.terasoluna.fw.batch.core.StandardBatchLogicExecutor	ビジネスロジック実行クラスの定義。	postProcessor	項番12を参照
20		batchUpdateProcessor	jp.terasoluna.fw.batch.standard.StandardBatchUpdateProcessor	バッチ更新処理クラスの定義。	throwableHandler	同期型起動用スレッドプール定義ファイルを参照 非同同期型起動用スレッドプール定義ファイルを参照
21	ジョブの処理状況クラス	MonitorableJobStatus	jp.terasoluna.fw.batch.monitor.MonitorableJobStatus	監視用ジョブの処理状況用Bean定義。	queueProcessor	項番14を参照
22	スレッドプールサイズ	threadSize	java.lang.Integer	スレッドプールのサイズ	queueLength	デフォルトBean定義ファイルを参照 ジョブBean定義ファイルを参照

ジョブ分割逐次単一トランザクションモデル

項番	設定項目名称	Bean ID	Classes/parent名	概要	property	property概要
0	ブレースホルダ			ブレースホルダの定義	import resource	初期値『classpath:common/PlaceholderConfig.xml』
1	親ジョブマネージャ	jobManager	baseManager (jp.terasoluna.fw.batch.core.JobManager)	親ジョブ実行用マネージャの定義。	name	初期値
					collector	初期値『sequentialSingleTransactionJobManager』
					workQueueFactory	ジョブBean定義ファイルを参照
					workProcessor	項番2を参照
					postProcessor	項番3を参照
					workerExecutorService	項番5を参照
2	親ジョブマネージャ	-	jp.terasoluna.fw.batch.common.JobManagerTaskWorkerQueueFactory	common)対応親ジョブ用作業キュー作成クラスの定義。	queueProcessor	同期型起動用スレッドプール定義ファイルを参照
					queueLength	非同期型起動用スレッドプール定義ファイルを参照
					supportProcessor	項番7を参照
3		parentJobPreProcessor	transactionalSupportProcessor (jp.terasoluna.fw.batch.spring.support.transactionalSupportProcessor)	トランザクション管理有りの親ジョブ前処理の定義。	supportProcessor	項番4を参照
4		-	standardSupportProcessor (jp.terasoluna.fw.batch.standard.StandardSupportProcessor)	親ジョブ前処理の定義。	supportLogicList	デフォルトBean定義ファイルを参照 ジョブBean定義ファイルを参照
5		parentJobPostProcessor	transactionalSupportProcessor (jp.terasoluna.fw.batch.spring.support.transactionalSupportProcessor)	トランザクション管理有りの親ジョブ後処理の定義。	supportProcessor	項番6を参照
6	親ジョブキュープロセッサ	-	standardSupportProcessor (jp.terasoluna.fw.batch.standard.StandardSupportProcessor)	親ジョブ後処理の定義。	supportLogicList	デフォルトBean定義ファイルを参照 ジョブBean定義ファイルを参照
7		sequentialSingleTransactionQueueProcessor	baseQueueProcessor (jp.terasoluna.fw.batch.standard.QueueProcessor)	親ジョブ用キュー処理クラスの定義。	name	初期値 『sequentialSingleTransactionQueueProcessor』
					worker	項番8を参照
8	子ジョブマネージャ	childJobManager	baseManager (jp.terasoluna.fw.batch.core.JobManager)	子ジョブ実行用マネージャの定義。	name	初期値 『childJobManager』
					collector	ジョブBean定義ファイルを参照
					workQueueFactory	項番9を参照
					preProcessor	項番10を参照
					postProcessor	項番12を参照
					workerExecutorService	同期型起動用スレッドプール定義ファイルを参照 非同期型起動用スレッドプール定義ファイルを参照
9	子ジョブマネージャ	-	jp.terasoluna.fw.batch.common.JobManagerTaskWorkerQueueFactory	common)対応子ジョブ用作業キュー作成クラスの定義。	queueProcessor	項番14を参照
					queueLength	デフォルトBean定義ファイルを参照
					supportProcessor	項番11を参照
10		jobPreProcessor	transactionalSupportProcessor (jp.terasoluna.fw.batch.spring.support.transactionalSupportProcessor)	トランザクション管理有りの子ジョブ前処理の定義。	supportLogicList	デフォルトBean定義ファイルを参照 ジョブBean定義ファイルを参照
11		-	standardSupportProcessor (jp.terasoluna.fw.batch.standard.StandardSupportProcessor)	子ジョブ前処理の定義。	supportProcessor	項番13を参照
12		jobPostProcessor	transactionalSupportProcessor (jp.terasoluna.fw.batch.spring.support.transactionalSupportProcessor)	トランザクション管理有りの子ジョブ後処理の定義。	supportLogicList	デフォルトBean定義ファイルを参照 ジョブBean定義ファイルを参照
13	子ジョブキュープロセッサ	-	standardSupportProcessor (jp.terasoluna.fw.batch.standard.StandardSupportProcessor)	子ジョブ後処理の定義。	supportLogicList	デフォルトBean定義ファイルを参照 ジョブBean定義ファイルを参照
					name	初期値 『singleTransactionChildQueueProcessor』
					worker	項番17を参照
					preProcessor	項番15を参照
					postProcessor	項番16を参照
					throwableHandler	ThrowableハンドラのBean定義。 初期設定値は『throwableHandler』である。 デフォルトBean定義ファイルを参照。
14	子ジョブキュープロセッサ	singleTransactionChildQueueProcessor	baseQueueProcessor (jp.terasoluna.fw.batch.standard.QueueProcessor)	子ジョブ用キュー処理クラスの定義。	supportLogicList	デフォルトBean定義ファイルを参照 ジョブBean定義ファイルを参照
15		firstChunkPreProcessor	standardSupportProcessor (jp.terasoluna.fw.batch.standard.StandardSupportProcessor)	先頭チャンク前処理の定義。		
16		lastChunkPostProcessor	standardSupportProcessor (jp.terasoluna.fw.batch.standard.StandardSupportProcessor)	最終チャンク後処理の定義。		
					batchUpdateProcessor	項番18を参照
					exceptionHandlerMap	デフォルトBean定義ファイルを参照
					defaultJobExceptionHandler	デフォルトBean定義ファイルを参照
17	ワーカ	singleTransactionWorker	jp.terasoluna.fw.batch.core.JobWorker	チャンク処理クラスの定義。	batchUpdateResultHandler	デフォルトBean定義ファイルを参照
					throwableHandler	ThrowableハンドラのBean定義。 初期設定値は『throwableHandler』である。 デフォルトBean定義ファイルを参照。
18		batchUpdateProcessor	jp.terasoluna.fw.batch.standard.StandardBatchUpdateProcessor	バッチ更新処理クラスの定義。	updateDAO	データアクセスBean定義ファイルを参照
19		batchUpdateProcessor	jp.terasoluna.fw.batch.core.StandardBatchUpdateProcessor	ビジネスロジック実行クラスの定義。	batchUpdateProcessor	ジョブBean定義ファイルを参照
					exceptionHandlerMap	デフォルトBean定義ファイルを参照
					defaultJobExceptionHandler	デフォルトBean定義ファイルを参照
20	ジョブの処理状況クラス	MonitorableJobStatus	jp.terasoluna.fw.batch.monitor.MonitorableJobStatus	監視用ジョブの処理状況用Bean定義。	throwableHandler	ThrowableハンドラのBean定義。 初期設定値は『throwableHandler』である。 デフォルトBean定義ファイルを参照。
					beanRegister	フレームワークBean定義ファイルを参照
					transactionManager	データアクセスBean定義ファイルを参照
					useSavepoint	デフォルトBean定義ファイルを参照
					partitionNo	初期値『-1』
					defaultJobExitCodeMap	デフォルトBean定義ファイルを参照
21	ジョブの処理状況クラス	JobStatus	jp.terasoluna.fw.batch.core.JobStatus	ジョブの処理状況用Bean定義。	partitionNo	初期値『-1』
					transactionManager	データアクセスBean定義ファイルを参照
					useSavepoint	デフォルトBean定義ファイルを参照
					partitionNo	初期値『-1』
					defaultJobExitCodeMap	デフォルトBean定義ファイルを参照
					constructor-arg	初期値『1』
22	分割ジョブの多重起動数	multiplicity	java.lang.Integer	分割ジョブの多重起動数		

## ファイルアクセス

項番	設定項目名称	Bean ID	Classes/parent名	概要	property	property概要
1	CSVファイルアクセス用（入力）DAO	csvFileQueryDAO	jp.terasoluna.fw.file.dao.standard.CSVFileQueryDAO / fileQueryDAO	CSVファイルアクセス用（入力）DAO	-	-
2	固定長ファイルアクセス用（入力）DAO	fixedFileQueryDAO	jp.terasoluna.fw.file.dao.standard.FixedFileQueryDAO / fileQueryDAO	固定長ファイルアクセス用（入力）DAO	-	-
3	可変長ファイルアクセス用（入力）DAO	variableFileQueryDAO	jp.terasoluna.fw.file.dao.standard.VariableFileQueryDAO / fileQueryDAO	可変長ファイルアクセス用（入力）DAO	-	-
4	その他ファイルアクセス用（入力）DAO	plainFileQueryDAO	jp.terasoluna.fw.file.dao.standard.PlainFileQueryDAO / fileQueryDAO	その他ファイルアクセス用（入力）DAO	-	-
5	ファイルアクセス用（入力）DAO 共通定義	fileQueryDAO	jp.terasoluna.fw.file.dao.standard.AbstractFileQueryDAO	ファイルアクセス用（入力）DAO 共通定義	textSetterMap	内容については項番6～9を参照
6		-	key: java.lang.String jp.terasoluna.fw.file.dao.standard.TextSetString	カラムフォーマットクラスの定義。（String型）	validator	項番10を参照
7		-	key: java.util.Date jp.terasoluna.fw.file.dao.standard.TextSetDate	カラムフォーマットクラスの定義。（Date型）	-	-
8		-	key: java.math.BigDecimal jp.terasoluna.fw.file.dao.standard.TextSetDecimal	カラムフォーマットクラスの定義。（BigDecimal型）	-	-
9		-	key: int jp.terasoluna.fw.file.dao.standard.TextSetInt	カラムフォーマットクラスの定義。（int型）	-	-
10		-	org.springframework.validation.commons.DefaultBeanValidator	バリデータのサンプルBean定義。	validatorFactory	項番11を参照
11		validatorFactory	jp.terasoluna.fw.validation.springmodules.DefaultValidatorFactoryEx	ファイル入力チェック（バリデータファクトリの記述例）	validationConfigLocations	-
12	CSVファイルアクセス用（出力）DAO	csvFileUpdateDAO	jp.terasoluna.fw.file.dao.standard.CSVFileUpdateDAO / fileUpdateDAO	CSVファイルアクセス用（出力）DAO	-	-
13	固定長ファイルアクセス用（出力）DAO	fixedFileUpdateDAO	jp.terasoluna.fw.file.dao.standard.FixedFileUpdateDAO / fileUpdateDAO	固定長ファイルアクセス用（出力）DAO	-	-
14	可変長ファイルアクセス用（出力）DAO	variableFileUpdateDAO	jp.terasoluna.fw.file.dao.standard.VariableFileUpdateDAO / fileUpdateDAO	可変長ファイルアクセス用（出力）DAO	-	-
15	その他ファイルアクセス用（出力）DAO	plainFileUpdateDAO	jp.terasoluna.fw.file.dao.standard.PlainFileUpdateDAO / fileUpdateDAO	その他ファイルアクセス用（出力）DAO	-	-
16	ファイルアクセス用（出力）DAO 共通定義	fileUpdateDAO	jp.terasoluna.fw.file.dao.standard.AbstractFileUpdateDAO	ファイルアクセス用（出力）DAO 共通定義	textGetterMap	内容については項番17～20を参照
17		-	key: java.lang.String jp.terasoluna.fw.file.dao.standard.TextGetString	ファイル書き込み用カラムフォーマットクラスの定義。（String型）	-	-
18		-	key: java.util.Date jp.terasoluna.fw.file.dao.standard.TextGetDate	ファイル書き込み用カラムフォーマットクラスの定義。（Date型）	-	-
19		-	key: java.math.BigDecimal jp.terasoluna.fw.file.dao.standard.TextGetDecimal	ファイル書き込み用カラムフォーマットクラスの定義。（BigDecimal型）	-	-
20		-	key: int jp.terasoluna.fw.file.dao.standard.TextGetInt	ファイル書き込み用カラムフォーマットクラスの定義。（int型）	-	-