

Magic xpi  
エラー処理  
コンセプト ペーパー



OUTPERFORM THE FUTURE™

The information in this manual/document is subject to change without prior notice and does not represent a commitment on the part of Magic Software Enterprises Ltd.

Magic Software Enterprises Ltd. makes no representations or warranties with respect to the contents hereof and specifically disclaims any implied warranties of merchantability or fitness for any particular purpose.

The software described in this document is furnished under a license agreement. The software may be used or copied only in accordance with the terms and conditions of the license agreement. It is against the law to copy the software on any medium except as specifically allowed in the license agreement.

No part of this manual and/or databases may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or information recording and retrieval systems, for any purpose other than the purchaser's personal use, without the prior express written permission of Magic Software Enterprises Ltd.

All references made to third-party trademarks are for informational purposes only regarding compatibility with the products of Magic Software Enterprises Ltd.

Unless otherwise noted, all names of companies, products, street addresses, and persons contained herein are part of a completely fictitious scenario or scenarios and are designed solely to document the use of Magic xpi.

Magic™ is a trademark of Magic Software Enterprises Ltd.

Btrieve® and Pervasive.SQL® are registered trademarks of Pervasive Software Inc.

IBM®, Topview™, System i5®/System i®/IBM i®, pSeries®, xSeries®, RISC System/6000®, DB2®, WebSphere®, Domino®, and Lotus Notes® are trademarks or registered trademarks of IBM Corporation.

Microsoft®, FrontPage®, Windows™, WindowsNT™, ActiveX™, Exchange™, Dynamics® AX, Dynamics® CRM, SharePoint®, Excel®, and Word® are trademarks or registered trademarks of Microsoft Corporation.

Oracle®, JD Edwards EnterpriseOne®, JD Edwards World®, and OC4J® are registered trademarks of the Oracle Corporation and/or its affiliates.

Google Calendar™ and Google Drive™ are trademarks of Google Inc.

Salesforce® is a registered trademark of salesforce.com Inc.

SAP® Business One and SAP® R/3® are registered trademarks of SAP AG in Germany and in several other countries.

SugarCRM is a trademark of SugarCRM in the United States, the European Union and other countries.

Linux® is a registered trademark of Linus Torvalds.

UNIX® is a registered trademark of UNIX System Laboratories.

GLOBEtrouter® and FLEXIm® are registered trademarks of Macrovision Corporation.

Solaris™ and Sun ONE™ are trademarks of Sun Microsystems Inc.

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

Red Hat® is a registered trademark of Red Hat Inc.

WebLogic® is a registered trademark of BEA Systems.

Interstage® is a registered trademark of the Fujitsu Software Corporation.

JBoss™ is a trademark of JBoss Inc.

Systinet™ is a trademark of Systinet Corporation.

GigaSpaces, GigaSpaces eXtreme Application Platform (XAP), GigaSpaces eXtreme Application Platform Enterprise Data Grid (XAP EDG), GigaSpaces Enterprise Application Grid, GigaSpaces Platform, and GigaSpaces, are trademarks or registered trademarks of GigaSpaces Technologies.

Clip art images copyright by Presentation Task Force®, a registered trademark of New Vision Technologies Inc.

This product uses the FreeImage open source image library. See <http://freeimage.sourceforge.net> for details

This product uses icons created by Axialis IconWorkShop™ (<http://www.axialis.com/free/icons>)

This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>).

This product includes software developed by Computing Services at Carnegie Mellon University (<http://www.cmu.edu/computing/>).

Copyright © 1989, 1991, 1992, 2001 Carnegie Mellon University. All rights reserved.

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (<http://www.openssl.org/>).

This product includes software that is Copyright © 1998, 1999, 2000 of the Thai Open Source Software Center Ltd. and Clark Cooper.

This product includes software that is Copyright © 2001-2002 of Networks Associates Technology Inc All rights reserved.

This product includes software that is Copyright © 2001-2002 of Cambridge Broadband Ltd. All rights reserved.

This product includes software that is Copyright © 1999-2001 of The OpenLDAP Foundation, Redwood City, California, USA. All Rights Reserved.

All other product names are trademarks or registered trademarks of their respective holders.

Copyright © 2016 by Magic Software Enterprises Ltd. All rights reserved.

## エラー処理の順序

Magic xpi は以下の順序でエラー処理を行います。:

1. ステップでエラーが発生した際、Magic xpi はエラー処理メカニズムに従い、ステップとトリガーの処理を停止します。  
注:データマップと SAP B1 アダプタのステップでは、処理を終了するか、継続するかを選択することができます。
2. エラーフローが設定されている場合は、そのフローを呼び出します。
3. エラーフローが設定されていない場合は、フローのエラーポリシーに従ってエラー処理を行います。
4. 次ステップの実行条件は、エラー変数 **C.sys.ErrorCode** に返された値を基に指定することができます。

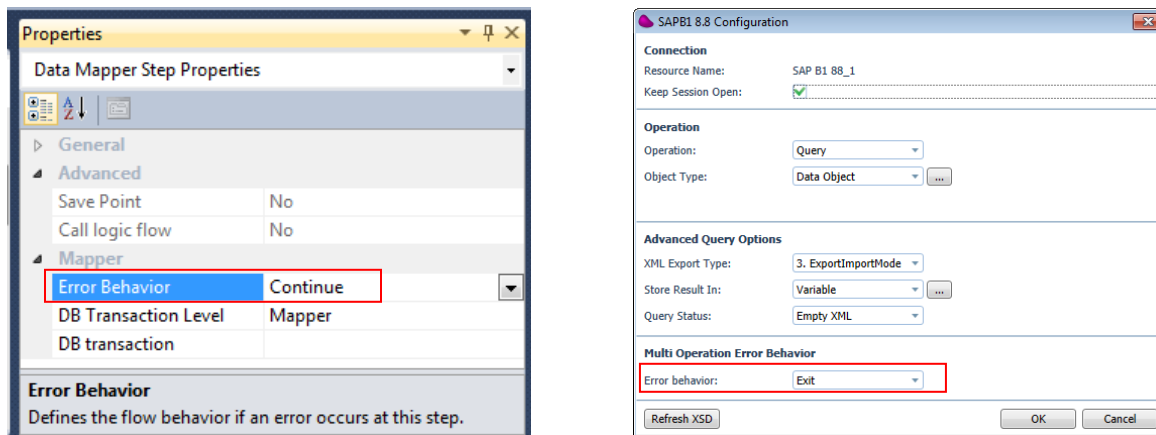
これらの手順の詳細を見てみましょう。

## 1-ステップでのエラー

ステップでエラーが発生すると、Magic xpi はステップを終了します。エラー変数 **C.sys.ErrorCode** はエラーコードで更新され、エラー処理メカニズムが働きます。(但し、データマップと SAP B1 ステップはエラー動作として **エラー動作** プロパティに **終了(Exit)** あるいは **継続(Continue)**を設定することができるため例外です。)

## エラー動作 プロパティ

データマップと SAP B1 ステップでは、エラー動作を選択することができます。

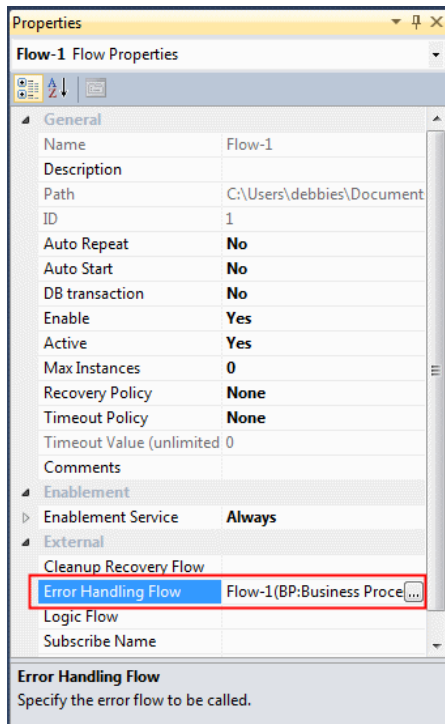


- **継続(Continue)**を選択した場合：エラーが発生するとエラーフローが呼び出され、エラーフローが終了すると元のフローに戻り、ステップの実行が継続されます。
- **終了(Exit)**を選択した場合：エラーが発生すると、ステップの実行は直ちに停止し、エラーが発生した時点からのマッピングやオペレーションは実行されません。但し、データマップにおいてエラー動作が**終了 (Exit)**、且つエラーフローが設定されている場合、エラー発生後に実行されるエラーフローでは（データマップ直後のステップではないため）エラー変数(C.sys.ErrorCode)はクリアされます。そのためデータマップはエラーがあるにも関わらず、レコードの処理を継続することになり、結果的に**終了 (Exit)** 設定を無視することになります。

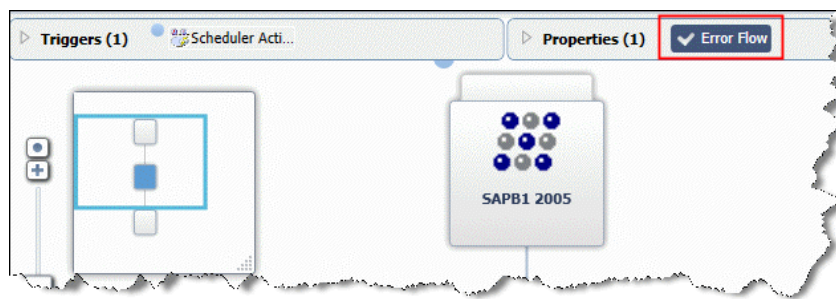
## 2- エラー処理フロー

エラーを処理するフローを作成できます。新しいフローを作成し、エラーを処理するための1つ以上のステップを定義するだけです。

このエラー処理フローは、フローのプロパティペインにある呼び出しフローのエラー処理フロープロパティを使用して、他のフローから使用することができます。



Error Handling Flow プロパティからフローを選択すると、下図に示すように、フロー領域の上部に



この指定により、元フローでエラーが発生すると、指定したエラーフローが起動されます。

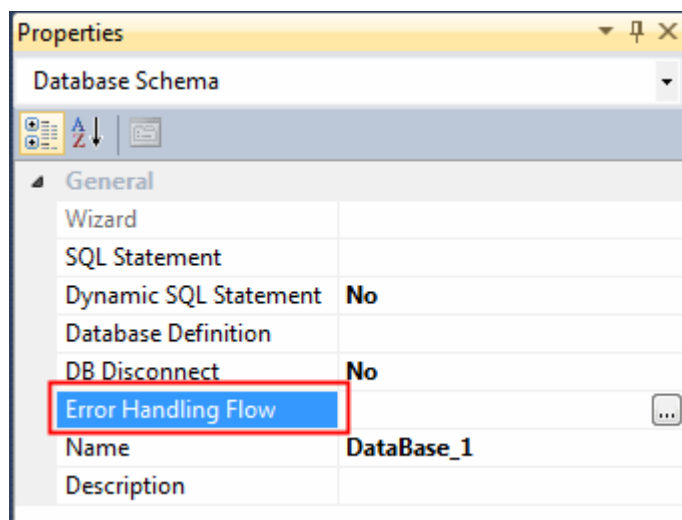
注: Magic xpi Server はエラーが発生すると、次のようなエラー関連情報を、エラー変数に自動的に更新します。エラーコード: C.sys.ErrorCode、エラー説明: C.sys.ErrorDescription、最終エラーコンポーネントタイプ: C.sys.LastErrorComponentType。標準では、C.sys.ErrorCode と C.sys.ErrorDescription はエラーフローが実行されるとクリアされます。エラーフロー内では、C.sys.LastErrorCode、C.sys.LastErrorDescription、C.sys.LastErrorInfo などの「Last」を含むエラー変数は、最後にエラーが発生

した時の値を保持しているので、エラーフロー内でエラー情報が必要な場合は、これらの変数を使用することができます。

## データマッパー – データベース送り先

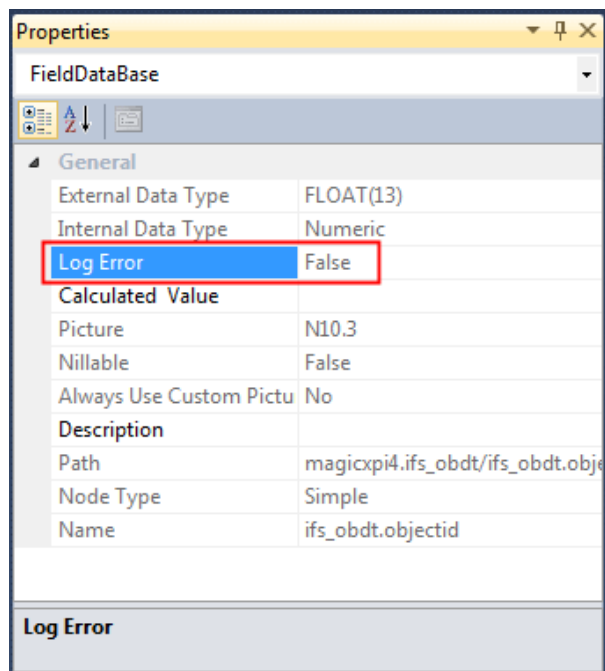
データマッパーの送り先がデータベースの場合、上記とは多少異なる動作になります。Magic xpi はまず最初にステップレベルのエラーフロー（データベーススキーマ プロパティ ペインの「エラー処理フロー」で指定）を呼び出そうとします。この指定により送り先毎にエラーフローを選択することができます。（送り先毎に異なるエラー処理フローを設定することができます）。

この設定によりエラーフローが呼び出される際に、元フローで失敗したステートメントから特定の項目の値をエラーフローのフロー変数に受け渡すことができます。



この場合、ノードプロパティダイアログボックスでは「ログエラー」が有効になり、この項目の値をフロー変数にセットできます。この変数の値はエラー発生時にエラー処理フローを呼び出す際に引き渡されます。

注: エラー処理フローに引き渡すフロー変数は、予め定義しておく必要があります。



## DB トランザクション レベル

データマップで送り先がデータベースの場合、DB トランザクションが設定されていると、エラー処理は以下のように実行されます。:

エラー動作	DB トランザクション レベル	動作
終了	マップ	全レコードがロールバックされます。
継続	マップ	レコードはロールバックされず、処理を継続します (エラーが発生した全ての接続先の該当レコードは除く)。
終了	レコード	エラー発生前の全レコードに対して更新処理が行われます。
継続	レコード	レコードはロールバックされず、処理を継続します (エラーが発生した全ての接続先の該当レコードは除く)。
終了	オートコミット	エラーが発生したレコードを除く全てのレコードが コミットされます。
継続	オートコミット	エラーが発生したレコードを除く全てのレコードがコ ミットされます。また、データマップは継続して次のレ コードを処理します。

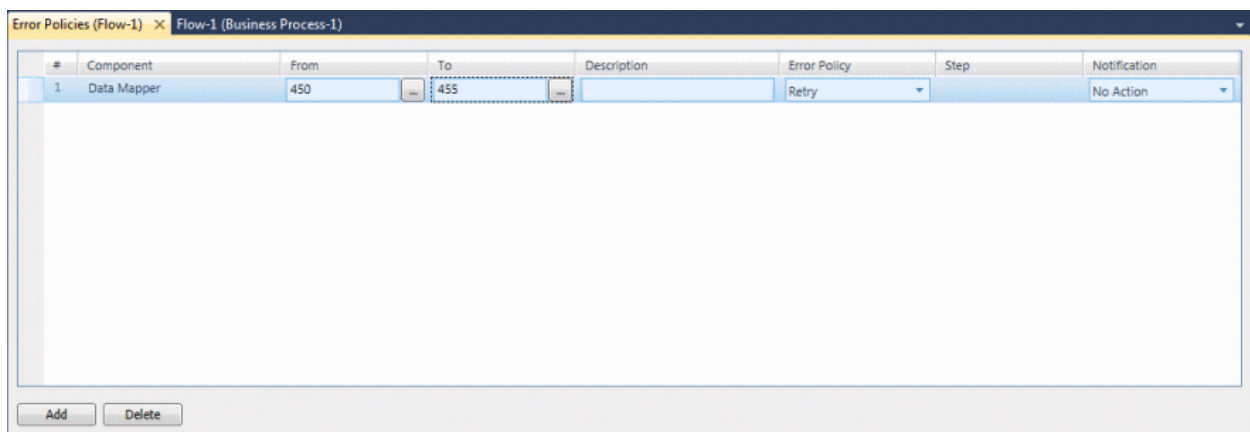
**注:** データベースを送り先に持つデータマップの処理中にエラーが発生した場合で、且つそのフローにエラーフローが設定されている場合（エラーがデータベースを含まないステップであっても）、エラーが発生するとエラーフローは直ちに呼び出されます。送り先へのデータの引き渡しは停止され、エラーフローが完了するまで一切のマッピングは行われません。この場合、エラーフロー中でログファイルを作成し、エラーコードとエラー説明を記録するようにして、それらの内容が失われないようすることができます。

### 3- エラーポリシー

エラーフローを設定しない場合、またはエラーフローの最後のステップで **C.sys.ErrorCode** や **C.sys.LastErrorComponentType** を更新している場合、Magic xpi はエラーが発生したフローのエラーポリシーを探し、それに従います。

エラーコードを更新する場合、その値は変数 **C.sys.LastErrorComponentType** に定義されたコンポーネントコードでなければいけません。エラーコードがコンポーネントに定義されていない値の場合、**エラー24**の一般エラーに属する「エラーコードは選択したコンポーネントに存在しません」という別のエラーが発生します。


エラーポリシーはフローのエラーポリシーリポジトリで定義します。上部バーのメニューから[フロー]>[エラー]を選択することで内容を確認することができます。

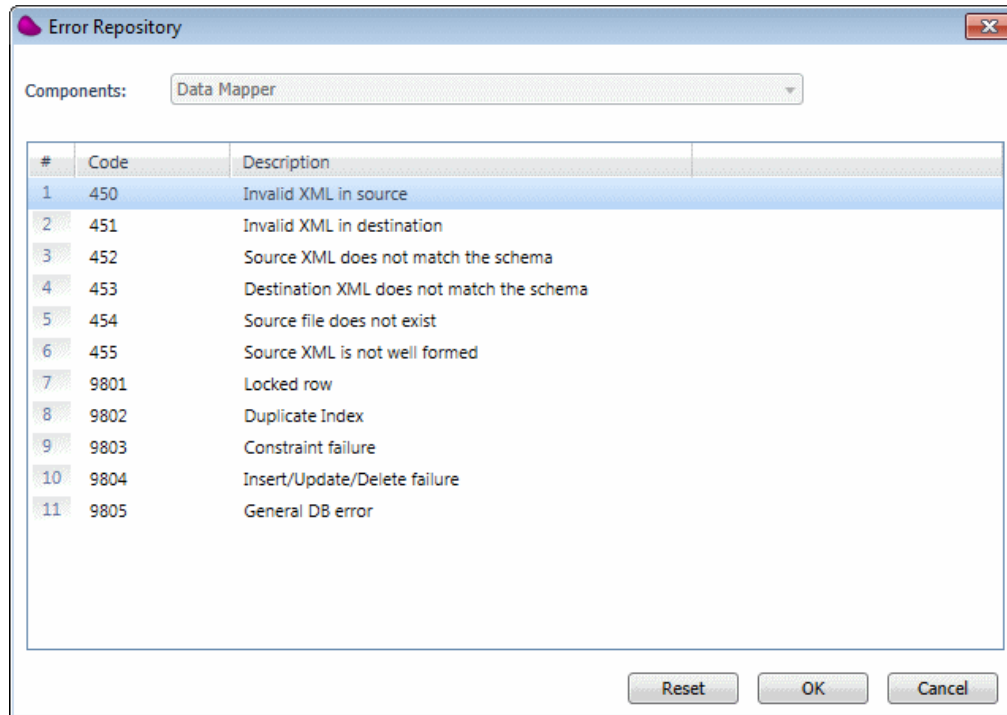


この画面で、エラーポリシー(アボート、無視、フロー再起動、リトライ、ジャンプ)を特定のエラー番号あるいはエラー番号の範囲で設定することができます。

エラーポリシーはフロー毎に設定することができます。例えばフロー1でエラーポリシーを「アボート」に設定し、フロー2では「無視」に設定することができます。

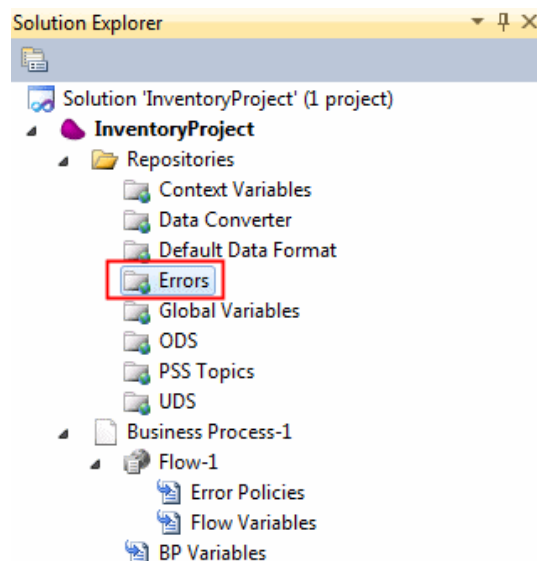


「From」または「To」の  ボタンをクリックすると、エラーポリシーリポジトリの内容が表示されます。この画面で Magic xpi に組み込まれている各コンポーネントのエラーコードを確認することができます。独自のコンポーネントを開発した場合は、このリストにエラーコードを登録する必要があります。

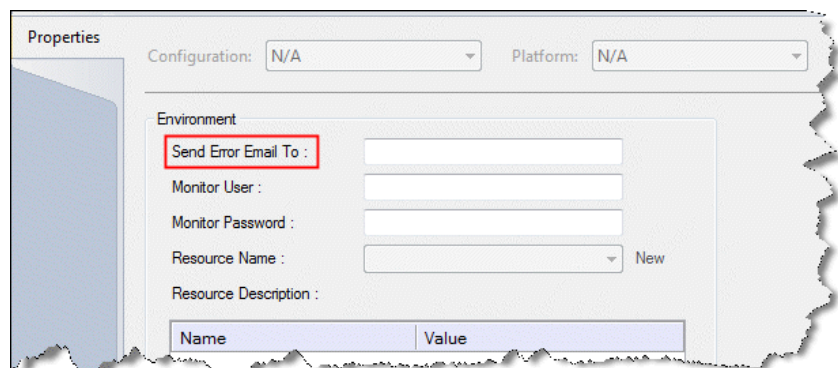


以下の手順で Magic xpi にエラーを追加します。:

1. From the 「ソリューションエクスプローラ」の「リポジトリ」セクションから「エラー」を選び、エラーリポジトリを開きます。
2. リポジトリ画面の上部にある「コンポーネント」ドロップダウンリストから「User Error」を選びます。
3. 「追加」ボタンをクリックします。



また、エラーポリシーに定義されている各エラー行で、管理者に対してエラー通知を行うように設定できます。（通知先メールアドレスは、プロジェクト特性ダイアログボックスの「エラーメール送信先」で設定します）。



The screenshot shows a 'Properties' dialog box with a 'Configuration' dropdown set to 'N/A' and a 'Platform' dropdown set to 'N/A'. The 'Environment' section is highlighted with a red box and contains the following fields:

- Send Error Email To :
- Monitor User :
- Monitor Password :
- Resource Name :  New
- Resource Description :

Below these fields is a table with two columns: 'Name' and 'Value'.

## エラーポリシーオプション

以下のオプションが用意されています。:

- **アボート** - 現在のフローを停止し、アボート処理を行います。
- **無視** - エラーの発生を無視し、フロー処理を継続します。
- **フロー再起動** - フローを最初のステップから再起動します。
- **リトライ** - 現在のステップをリトライします。IFS.ini ファイル中のパラメータ「StepRetryLimit=フラグ」でリトライする回数を指定することができます。デフォルトは3回です。「StepRetryDelayTnSec=フラグ」を使用して再試行の遅延時間を制御することができます。Magic xpi サーバがステップを実行することなく、限界に達した場合、最後のエラーは「c.sys.ErrorCode」コンテキスト変数に保存されます。「StepRetryClearError=フラグ」を使用して変数からエラーがクリアされるかどうかを制御することができます。
- **ジャンプ** - 他のステップ（フローエラーリポジトリの**ステップ列**に表示）にジャンプします。ジャンプを使用することで他のステップに処理を飛ばすことができます。フロー内に実行条件が False の分岐を作成しておき、ジャンプを使用した時のみその分岐を実行させることができます。ジャンプを使用すると、フローはそのステップから処理が継続され、エラーが発生したステップ以降の処理は継続されません。

**注:** Magic xpi にはアボートフロー ユーティリティも用意されています。このサービスを使用するとフローロジックの一部として意図的にフローの実行をアボートさせることができます。たアボートフローサービスにはアボート後、フローを自動的に再起動するオプションも準備されています。

フローがアボートした後、クリーンアップリカバリーフローが実行されます。(存在する場合)。

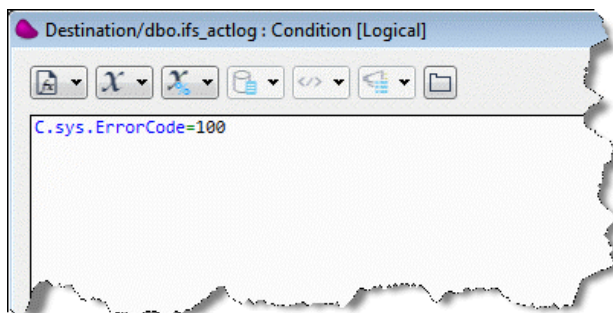
詳細は Magic xpi ヘルプを参照してください。

## 4-ステップでのエラー処理

ステップの実行条件でエラーコードをチェックすることでエラー処理を行うこともできます。直前のステップでエラーが発生すると、変数 `C.sys.ErrorCode` にエラーコードが設定され、次ステップの条件判断時までその値が保持されています。この値は次ステップの処理開始と同時にクリアされます。

特定のステップでエラーをキャッチし、処理をしたい場合、次ステップの実行条件でエラー変数を使用します。

エラーコードの値はエラーリポジトリで確認することができます。(ソリューションエクスプローラ リポジトリ セクション> エラー)。



## エラー変数

Magic xpi にはプロジェクト内のフローのエラー処理に使用できる変数が用意されています。

a. 次の変数はステップでの処理が完了すると、値がクリアされます。:

- **C.sys.ErrorCode** –サーバでステップが実行される度に、ステップにエラーが発生すると、変数 `C.sys.ErrorCode` にはエラーコードが設定されます。この変数の値はステップの処理が完了するとクリアされます（次ステップの条件判定では使用が可能です）。
- **C.sys.ErrorDescription** –サーバでステップが実行される度に、ステップにエラーが発生すると、変数 `C.sys.ErrorDescription` にはエラー内容（説明）が設定されます。この変数の値はステップの処理が完了するとクリアされます。（次ステップの条件判定では使用が可能です）。

b. 以下の変数（Last 付変数）はステップの処理が完了しても値はクリアされません。新たなエラーが発生すると、エラー内容が更新されます。

- **C.sys.LastErrorFlowName** – サーバでステップが実行される度、ステップにエラーが発生すると C.sys.LastErrorFlowName 変数にはエラーが発生したフローの名称が設定されます。
- **C.sys.LastErrorStepName** – サーバでステップが実行される度、ステップにエラーが発生すると C.sys.LastErrorStepName 変数にはエラーが発生したステップの名称が設定されます。
- **C.sys.LastErrorInfo** – エラーに関する詳細情報が設定されます(例：SQL 文)。
- **C.sys.LastErrorCode** – サーバでステップが実行される度、ステップにエラーが発生すると C.sys.LastErrorCode 変数にはエラーコードが設定されます。
- **C.sys.LastErrorDescription** – サーバでステップが実行される度、ステップにエラーが発生すると C.sys.LastErrorDescription 変数にはエラー内容（説明）が設定されます。
- **C.sys.LastErrorComponentType** –サーバでステップが実行される度、ステップにエラーが発生すると C.sys.LastErrorComponentType 変数にはエラーが発生したコンポーネントタイプ（コンポーネント ID）が設定されます。コンポーネント ID はコンポーネントリポジトリ（プロジェクト メニュー>コンポーネント）で確認することができます。

上記の変数はコンテキスト変数であり、グローバル変数ではありません。従って、フローが並列で実行されている場合、その値は各フロー内で異なります。

各コンポーネントには個別のエラーコードが設定されています。エラーを発生させたい場合、**C.sys.ErrorCode** のエラー番号と **C.sys.LastErrorComponentType** に関連するコンポーネントタイプ（コンポーネント ID）を必ず更新しなければなりません。

## エラー処理に関する補足事項

### データベーストランザクションとエラー処理

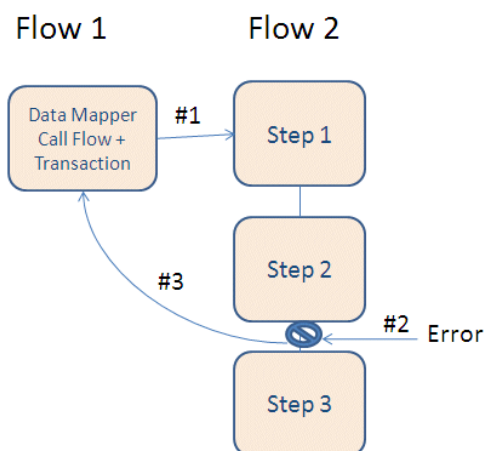
フロー特性の DB トランザクション を Yes に設定すると、フロー内に含まれるデータベースに接続するすべてのデータマッピングステップが同一のトランザクションになります。その際、フロー特性の DB トランザクションでは、フロー内で使用しているすべてのデータベース(リソース)を選択しなければなりません。同一コンテキスト内で特定のデータベースのみをトランザクションから除外するオプションはありません。特定のデータマッピングステップをトランザクションから除外したい場合は、除外したいステップをパラレルステップとして実行させなければなりません。

- 各データマッピングステップのエラー動作が**継続(Continue)**に設定されている場合、1 レコードのみ、ロールバックされます；後続の処理は継続されます。しかしながらフローの実行時にクラッシュした場合、トランザクション内のすべてのデータベース オペレーションはロールバックされます。
- 各データマッピングステップのエラー動作が**終了(Exit)**に設定されている場合、トランザクション内のすべてのデータベース オペレーションはロールバックされ、フローは停止します。さらに親フローにより呼び出される場合、呼び出し元フローの実行も停止します。

### コールフローでのエラー処理

#### データベースエラー

以下の図のように、あるフロー(Flow 1)でトランザクションがオープンされており、それか他のフロー(Flow 2)を呼び出す場合、トランザクションは Flow 2 全体で有効となります。それで、Flow 2 でエラーが発生した場合、Magic xpi は Flow 2 をアボートし、Flow 1 に戻ります。



データマッピングの DB トランザクション レベルがマッピングに設定されている場合、トランザクションはデータマッピングが開始する前にオープンされます。そのため、呼び出されたフローでエラーが発生した場合、データマッピングはエラーで処理を終了します。データマッピングのエラー動作で**終了(Exit)**あるいは**継続(Continue)**が選択されていても、ステップよりも上位のレベルでトランザクションがオープンされているため、このエラー動作は無視されます。

## 非データベースエラー

送り元で Flat File(フラットファイル)、送り先で Call Flow を設定し、呼び出されたフローが非データベースエラー(例: アボート)で終了した場合、データマッパーはエラー動作で指定された内容に従い、以降の処理を行います。エラー発生時の動作は、終了するかフラットファイルの次のレコードを処理するかになります。エラー動作に継続(Continue)が選択されている場合は、呼び出されたフローがエラーで終了しているので、データマッパーはエラーフローを呼び出しません。終了(Exit)が選択されている場合は、データマッパーは呼び出されたフローがエラーなので、それによって処理を終了します。それ以降は Magic xpi の標準的なエラー処理動作に従います。

## トリガーでのエラー管理

トリガーでエラーが発生すると、フロー特性で指定するエラー フローが呼び出されます。

C.sys.InvokingComponentName 変数にはトリガー名が設定され、C.sys.InvokingFlowName 変数にはトリガーにより起動されたフロー名が設定されます。

## XML インターフェイスでの複数オペレーション

データマッパーは複数オペレーション(例: 複数レコードの更新)をサポートしています。

Salesforce や SharePoint のような XML インターフェイスでの複数オペレーションをサポートしているコンポーネントでは、コンポーネントから返されるエラーは最終オペレーションの情報です。複数オペレーションを実行する際は 処理成功 で設定する変数のチェックに頼るのではなく、戻り XML の内容をチェック(XML を解析し、個々に判断)する必要があります。

これらのコンポーネントでは、エラーは戻り XML の一部(XML 内にエラー情報が存在する)であるため、各オペレーション毎にエラーは発生しません。エラーチェックを行うためには、XML を確認(XML を解析する)しなければなりません。C.sys.LastErrorInfo 変数にはエラー情報は設定されません。

## モニタに表示するエラーメッセージの制御

[MAGIC\_IBOLT]MonitorLogLevel =\* All

Magic xpi の ifs.ini ファイル内のこのフラグを設定することでモニタにエラーメッセージを出力するか否かを設定することができます。このフラグの値は、以下のどれかになります。カンマ区切りで複数設定することができます。:

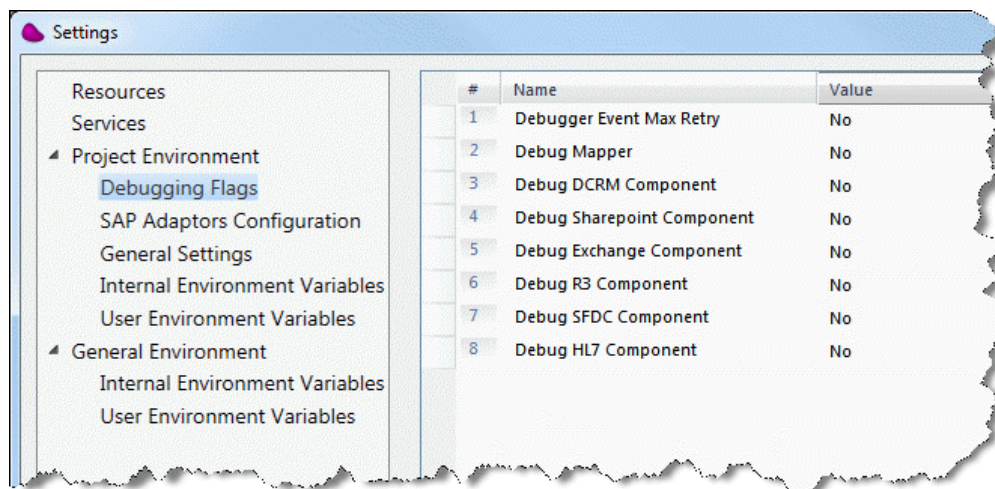
- **User** – ログとユーザー定義メッセージのみを記録し表示します。
- **Error** – エラーを表示し、エラーコンポーネントを呼び出します。
- **Exec** – サーバの起動やフローの実行開始などの実行関連のメッセージを表示します。このスイッチには、エラーとユーザメッセージが含まれています。
- **Service** – ODS やロック、および PSS などのサービス関連のメッセージを表示します。
- **All** – すべてのメッセージを表示します。(フラグが存在しない場合、デフォルトになります。)
- **None** – メッセージを表示しません。

注: このフラグに複数の値を定義する場合、(上記のように) = の後にアスタリスク (\*) を入力してください。

## データマッパーと各種コネクタのデバッグ

データマッパーと各種コネクタをデバッグするには:

Data Mapper と各種コネクタをデバッグし、DBMS から受信した SQL 文に関するエラー情報を取得するには、**設定ダイアログボックス (プロジェクトメニュー>設定>プロジェクト環境>デバッグフラグ)** でプロジェクトの設定を使用します。



Magic xpi はログファイルを Magic xpi の logs フォルダに作成します。

## SAP R/3 コネクタのデバッグ:

SAP R/3 コネクタをデバッグするには、<Magic xpi installation>/java/classes フォルダ配下にある log4j.xml ファイルに定義されているデバッグ レベルを変更する必要があります。

Magic xpi はログファイルを Magic xpi logs フォルダに作成します。

log4j.xml ファイル内の以下の2つのセクションは SAP R/3 のステップでのデバッグとトリガーでのデバッグレベルを定義しています。SAP R/3 コネクタをデバッグするには、赤字で示されている level value を off から debug (<level value="debug"/>)に変更しなければなりません。

### SAP R/3 ステップ デバッグ

```
<!--
sapr3 ifc logger:
should not pass messages up the hierarchy (additivity=false)
-->
<logger name="com.magicsoftware.ibolt.sapr3.IBSapR3" additivity="false">
    <level value="off"/>
    <appender-ref ref="ifc.sapr3"/>
</logger>
```

### SAP R/3 トリガー デバッグ

```
<!--
sapr3 Trigger logger:
should not pass messages up the hierarchy (additivity=false)
-->
<logger name="com.magicsoftware.ibolt.sapr3.IBSapR3Server" additivity="false">
    <level value="off"/>
    <appender-ref ref="ifc.sapr3Trigger"/>
</logger>
```



## UDS とデータベースを組み合わせた場合のエラー

UDS とデータベースの組合せでは、送り先の定義順序で動作が決まります。

- UDS がデータベースより先に定義されている場合でデータベースでエラーが発生した場合、UDS にはレコードが追加されません。
- データベースが先に定義されている場合:
  - エラー動作が 終了(Exit) の場合、レコードは UDS に追加されません。
  - エラー動作が 継続(Continue)、レコードは UDS に追加されます。

この動作は UDS とデータベースの組み合わせに限定される動作です。たとえば、送り先の組み合わせがデータベースとフラット ファイルだった場合、データベースの定義が最初に行われている場合でも、レコードは常にフラット ファイルに追加されます。

### .NET ユーティリティ - エラー処理

以下の手順で .NET ユーティリティのエラーコードを Magic xpi に新しく追加することができます。:

1. ソリューションエクスプローラのリポジトリセクションから、エラーを選択し、エラーリポジトリ画面を開きます。
2. リポジトリの上部にあるコンポーネント ドロップダウンリストより、User Error を選択します。
3. 新規ボタンをクリックしてエラーを追加します。

.NET クラスの Invoke メソッドの IBFlowContext オブジェクトの ErrorCode および ErrorMessage プロパティに値を設定することでエラーを設定することができます。ErrorCode プロパティの値は、エラーリポジトリに予め .NET ユーティリティのエラーとして登録されている値を設定しなければなりません。このリポジトリ内で、コンポーネントのドロップダウンリストから .NET オプションを選択することにより、.NET ユーティリティのエラーを絞り込むことができます。

エラーポリシーリポジトリで .NET ステップで発生したエラーのエラー動作を定義することができます。

コンテキスト変数 C.sys.ErrorCode と C.sys.ErrorDescription は以下の手順で更新されます:

1. ErrorCode と ErrorMessage の両方を .NET 側で更新すると、Magic xpi は上記 2 つのコンテキスト変数を .NET 側で更新された値で更新します。
2. .NET 側で ErrorCode を更新し、ErrorMessage を更新しなかった場合、Magic xpi は C.sys.ErrorCode 変数を .NET 側の ErrorCode で更新し、エラー リポジトリにその ErrorCode が定義されている場合、C.sys.ErrorDescription 変数はエラー リポジトリに定義されている値で更新されます。
3. 例外が発生した場合、Magic xpi は C.sys.ErrorCode 変数に 777 を設定し、C.sys.ErrorDescription 変数には .NET 側で発生した例外が設定されます。

4. **ErrorCode** がエラー リポジトリのユーザ エラー リストに定義されていない場合、**Error Code does not exist for the selected component** (選択したコンポーネントにはエラー コードが存在しない) というエラーが返されます。

## サンプルプロジェクト

Magic xpi には、Magic xpi がエラーフローを処理する方法を示すサンプルプロジェクトが含まれています。 サンプルプロジェクトは<Magic xpi installation>\Extra\SampleProjects にあります。