



Informatica® Cloud Application Integration  
October 2023

**デザイン**

© 著作権 Informatica LLC 1993, 2023

本ソフトウェアおよびマニュアルには、Informatica LLC の所有権下にある情報が収められています。これらは使用および開示の制限等を定めた使用許諾契約のもとに提供され、著作権法により保護されています。当該ソフトウェアのリバースエンジニアリングは禁じられています。本マニュアルのいかなる部分も、いかなる手段（電子的複製、写真複製、録音など）によっても、Informatica LLC の事前の承諾なしに複製または転載することは禁じられています。このソフトウェアは、米国および/または国際的な特許、およびその他の出願中の特許によって保護されています。

合衆国政府によるソフトウェアの使用、複製または開示は、DFARS 227.7202-1 (a) および 227.7702-3 (a) (1995 年)、DFARS 252.227-7013(C) (1) (ii) (1988 年 10 月)、FAR 12.212 (a) (1995 年)、FAR 52.227-19、または FAR 52.227-14 (ALT III) に記載されているとおり、当該ソフトウェア使用許諾契約に定められた制限によって規制されます。

本製品または本書の情報は、予告なしに変更されることがあります。お客様が本製品または本書内に問題を発見された場合は、書面に当社までお知らせください。

Informatica、Informatica Platform、Informatica Data Services、PowerCenter、PowerCenterRT、PowerCenter Connect、PowerCenter Data Analyzer、PowerExchange、PowerMart、Metadata Manager、Informatica Data Quality、Informatica Data Explorer、Informatica B2B Data Transformation、Informatica B2B Data Exchange、Informatica On Demand、Informatica Identity Resolution、Informatica Application Information Lifecycle Management、Informatica Complex Event Processing、Ultra Messaging、Informatica Master Data Management、および Live Data Map は、Informatica LLC の米国および世界中の管轄地での商標または登録商標です。その他のすべての企業名および製品名は、それぞれの企業の商標または登録商標です。

本ソフトウェアまたはドキュメントの一部は、次のサードパーティが有する著作権に従います（ただし、これらに限定されません）。Copyright DataDirect Technologies. All rights reserved. Copyright (C) Sun Microsystems. All rights reserved. Copyright (C) RSA Security Inc. All rights reserved. Copyright (C) Ordinal Technology Corp. All rights reserved. Copyright (C) Aandacht c.v. All rights reserved. Copyright Genivia, Inc. All rights reserved. Copyright Isomorphic Software. All rights reserved. Copyright (C) Meta Integration Technology, Inc. All rights reserved. Copyright (C) Intalio. All rights reserved. Copyright (C) Oracle. All rights reserved. Copyright (C) Adobe Systems Incorporated. All rights reserved. Copyright (C) DataArt, Inc. All rights reserved. Copyright (C) ComponentSource. All rights reserved. Copyright (C) Microsoft Corporation. All rights reserved. Copyright (C) Rogue Wave Software, Inc. All rights reserved. Copyright (C) Teradata Corporation. All rights reserved. Copyright (C) Yahoo! Inc. All rights reserved. Copyright (C) Glyph & Cog, LLC. All rights reserved. Copyright (C) Thinkmap, Inc. All rights reserved. Copyright (C) Clearpace Software Limited. All rights reserved. Copyright (C) Information Builders, Inc. All rights reserved. Copyright (C) OSS Nokalva, Inc. All rights reserved. Copyright Edifecs, Inc. All rights reserved. Copyright Cleo Communications, Inc. All rights reserved. Copyright (C) International Organization for Standardization 1986. All rights reserved. Copyright (C) ej-technologies GmbH. All rights reserved. Copyright (C) Jaspersoft Corporation. All rights reserved. Copyright (C) International Business Machines Corporation. All rights reserved. Copyright (C) yWorks GmbH. All rights reserved. Copyright (C) Lucent Technologies. All rights reserved. Copyright (C) University of Toronto. All rights reserved. Copyright (C) Daniel Veillard. All rights reserved. Copyright (C) Unicode, Inc. Copyright IBM Corp. All rights reserved. Copyright (C) MicroQuill Software Publishing, Inc. All rights reserved. Copyright (C) PassMark Software Pty Ltd. All rights reserved. Copyright (C) LogiXML, Inc. All rights reserved. Copyright (C) 2003-2010 Lorenzi Davide, All rights reserved. Copyright (C) Red Hat, Inc. All rights reserved. Copyright (C) The Board of Trustees of the Leland Stanford Junior University. All rights reserved. Copyright (C) EMC Corporation. All rights reserved. Copyright (C) Flexera Software. All rights reserved. Copyright (C) Jinfonet Software. All rights reserved. Copyright (C) Apple Inc. All rights reserved. Copyright (C) Telerik Inc. All rights reserved. Copyright (C) BEA Systems. All rights reserved. Copyright (C) PDFlib GmbH. All rights reserved. Copyright (C) Orientation in Objects GmbH. All rights reserved. Copyright (C) Tanuki Software, Ltd. All rights reserved. Copyright (C) Ricebridge. All rights reserved. Copyright (C) Sencha, Inc. All rights reserved. Copyright (C) Scalable Systems, Inc. All rights reserved. Copyright (C) jQWidgets. All rights reserved. Copyright (C) Tableau Software, Inc. All rights reserved. Copyright (C) MaxMind, Inc. All rights reserved. Copyright (C) TMat Software s.r.o. All rights reserved. Copyright (C) MapR Technologies Inc. All rights reserved. Copyright (C) Amazon Corporate LLC. All rights reserved. Copyright (C) Highsoft. All rights reserved. Copyright (C) Python Software Foundation. All rights reserved. Copyright (C) BeOpen.com. All rights reserved. Copyright (C) CNRI. All rights reserved.

本製品には、Apache Software Foundation (<http://www.apache.org/>) によって開発されたソフトウェア、およびさまざまなバージョンの Apache License（まとめて「License」と呼んでいます）の下に許諾された他のソフトウェアが含まれます。これらのライセンスのコピーは、<http://www.apache.org/licenses/> で入手できます。適用法にて要求されないが書面に同意されない限り、ライセンスの下に配布されるソフトウェアは「現状のまま」で配布され、明示的あるいは黙示的かを問わず、いかなる種類の保証や条件も付帯することはありません。ライセンス下での許諾および制限を定める具体的文言については、ライセンスを参照してください。

本製品には、Mozilla (<http://www.mozilla.org/>) によって開発されたソフトウェア、ソフトウェア Copyright (c) The JBoss Group, LLC, all rights reserved、ソフトウェア Copyright (c) 1999-2006 by Bruno Lowagie and Paulo Soares および GNU Lesser General Public License Agreement のさまざまなバージョン (<http://www.gnu.org/licenses/lgpl.html> で参照できる場合がある) に基づいて許諾されたその他のソフトウェアが含まれています。資料は、Informatica が無料で提供しており、一切の保証を伴わない「現状渡し」で提供されるものとし、Informatica LLC は市場性および特定の目的の適合性の黙示の保証などを含めて、一切の明示的及び黙示的保証の責任を負いません。

製品には、ワシントン大学、カリフォルニア大学アーバイン校、およびバンダービルト大学の Douglas C. Schmidt および同氏のリサーチグループが著作権を持つ ACE (TM) および TAO (TM) ソフトウェアが含まれています。Copyright (C) 1993-2006, All rights reserved.

本製品には、OpenSSL Toolkit を使用するために OpenSSL Project が開発したソフトウェア (copyright The OpenSSL Project. All Rights Reserved) が含まれています。また、このソフトウェアの再配布は、<http://www.openssl.org> および <http://www.openssl.org/source/license.html> にある使用条件に従います。

本製品には、Curl ソフトウェア Copyright 1996-2013, Daniel Stenberg, <daniel@haxx.se> が含まれます。All rights reserved. 本ソフトウェアに関する許諾および制限は、<http://curl.haxx.se/docs/copyright.html> にある使用条件に従います。すべてのコピーに上記の著作権情報とこの許諾情報が記載されている場合、目的に応じて、本ソフトウェアの使用、コピー、変更、ならびに配布が有償または無償で許可されます。

本製品には、MetaStuff, Ltd. のソフトウェアが含まれます。Copyright 2001-2005 (C) MetaStuff, Ltd. All Rights Reserved. 本ソフトウェアに関する許諾および制限は、<http://www.dom4j.org/license.html> にある使用条件に従います。

製品には、The Dojo Foundation のソフトウェアが含まれます。Copyright (C) 2004-2007. All rights reserved. 本ソフトウェアに関する許諾および制限は、<http://dojotoolkit.org/license> にある使用条件に従います。

本製品には、ICU ソフトウェアおよび他のソフトウェアが含まれます。Copyright International Business Machines Corporation. All rights reserved. 本ソフトウェアに関する許諾および制限は、<http://source.icu-project.org/repos/icu/icu/trunk/license.html> にある使用条件に従います。

本製品には、Per Bothner のソフトウェアが含まれます。Copyright (C) 1996-2006. All rights reserved. お客様がこのようなソフトウェアを使用するための権利は、ライセンスで規定されています。<http://www.gnu.org/software/kawa/Software-License.html> を参照してください。

本製品には、OSSP UUID ソフトウェアが含まれます。Copyright (C) 2002 Ralf S. Engelschall, Copyright (C) 2002 The OSSP Project Copyright (C) 2002 Cable & Wireless Deutschland. 本ソフトウェアに関する許諾および制限は、<http://www.opensource.org/licenses/mit-license.php> にある使用条件に従います。

本製品には、Boost (<http://www.boost.org/>) によって開発されたソフトウェア、または Boost ソフトウェアライセンスの下で開発されたソフトウェアが含まれます。本ソフトウェアに関する許諾および制限は、[http://www.boost.org/LICENSE\\_1\\_0.txt](http://www.boost.org/LICENSE_1_0.txt) にある使用条件に従います。

本製品には、University of Cambridge のが含まれます。Copyright (C) 1997-2007. 本ソフトウェアに関する許諾および制限は、<http://www.pcre.org/license.txt> にある使用条件に従います。

本製品には、The Eclipse Foundation のソフトウェアが含まれます。Copyright (C) 2007. All rights reserved. 本ソフトウェアに関する許諾および制限は、<http://www.eclipse.org/org/documents/epl-v10.php> および <http://www.eclipse.org/org/documents/edl-v10.php> にある使用条件に従います。

本製品には、<http://www.tcl.tk/software/tcltk/license.html>、<http://www.bosrup.com/web/overlib/?License>、<http://www.stlport.org/doc/license.html>、<http://www.asm.ow2.org/license.html>、<http://www.cryptix.org/LICENSE.TXT>、<http://hsqldb.org/web/hsqldbLicense.html>、<http://httpunit.sourceforge.net/doc/license.html>、<http://jung.sourceforge.net/license.txt>、[http://www.gzip.org/zlib/zlib\\_license.html](http://www.gzip.org/zlib/zlib_license.html)、<http://www.openldap.org/software/release/license.html>、<http://www.libssh2.org>、<http://sf4j.org/license.html>、<http://www.sente.ch/software/OpenSourceLicense.html>、<http://fusesource.com/downloads/license-agreements/fuse-message-broker-v-5-3-license-agreement>、<http://antlr.org/license.html>、<http://aopalliance.sourceforge.net/>、<http://www.bouncycastle.org/license.html>、<http://www.jgraph.com/jgraphdownload.html>、<http://www.jcraft.com/jsch/LICENSE.txt>、[http://jotm.objectweb.org/bsd\\_license.html](http://jotm.objectweb.org/bsd_license.html) に基づいて許諾されたソフトウェアが含まれています。<http://www.w3.org/Consortium/Legal/2002/copyright-software-20021231>、<http://www.sf4j.org/license.html>、<http://nanoxml.sourceforge.net/orig/copyright.html>、<http://www.json.org/license.html>、<http://forge.ow2.org/projects/javaservice/>、<http://www.postgresql.org/about/licence.html>、<http://www.sqlite.org/copyright.html>、<http://www.tcl.tk/software/tcltk/license.html>、<http://www.jaxen.org/faq.html>、<http://www.jdom.org/docs/faq.html>、<http://www.sf4j.org/license.html>、<http://www.iodbc.org/dataspace/iodbc/wiki/iODBC/License>、<http://www.keplerproject.org/md5/license.html>、<http://www.toedter.com/en/jcalendar/license.html>、<http://www.edankert.com/bounce/index.html>、<http://www.net-snmp.org/about/license.html>、<http://www.openmdx.org/#FAQ>、[http://www.php.net/license/3\\_01.txt](http://www.php.net/license/3_01.txt)、<http://srp.stanford.edu/license.txt>、<http://www.schneier.com/blowfish.html>、<http://www.jmock.org/license.html>、<http://xsom.java.net>、<http://benalman.com/about/license/>、<https://github.com/CreateJS/EaselJS/blob/master/src/easeljs/display/Bitmap.js>、<http://www.h2database.com/html/license.html#summary>、<http://jsoncpp.sourceforge.net/LICENSE>、<http://jdbc.postgresql.org/license.html>、<http://protobuf.googlecode.com/svn/trunk/src/google/protobuf/descriptor.proto>、<https://github.com/rantav/hector/blob/master/LICENSE>、<http://web.mit.edu/Kerberos/krb5-current/doc/mitK5license.html>、<http://jibx.sourceforge.net/jibx-license.html>、<https://github.com/lyokato/libgeohash/blob/master/LICENSE>、<https://github.com/hjiang/jsonxx/blob/master/LICENSE>、<https://code.google.com/p/lz4/>、<https://github.com/jedisct1/libodium/blob/master/LICENSE>、<http://one-jar.sourceforge.net/index.php?page=documents&file=license>、<https://github.com/EsotericSoftware/kryo/blob/master/license.txt>、<http://www.scala-lang.org/license.html>、<https://github.com/tinkerpop/blueprints/blob/master/LICENSE.txt>、<http://gee.cs.oswego.edu/dl/classes/EDU/oswego/cs/dl/util/concurrent/intro.html>、<https://aws.amazon.com/asi/>、<https://github.com/twbs/bootstrap/blob/master/LICENSE>、および <https://sourceforge.net/p/xmlunit/code/HEAD/tree/trunk/LICENSE.txt>。

本製品には、Academic Free License (<http://www.opensource.org/licenses/afl-3.0.php>)、Common Development and Distribution License (<http://www.opensource.org/licenses/cddl1.php>)、Common Public License (<http://www.opensource.org/licenses/cpl1.0.php>)、Sun Binary Code License Agreement Supplemental License Terms、BSD License (<http://www.opensource.org/licenses/bsd-license.php>)、BSD License (<http://opensource.org/licenses/BSD-3-Clause>)、MIT License (<http://www.opensource.org/licenses/mit-license.php>)、Artistic License (<http://www.opensource.org/licenses/artistic-license-1.0>)、Initial Developer's Public License Version 1.0 (<http://www.firebirdsql.org/en/initial-developer-s-public-license-version-1-0/>) に基づいて許諾されたソフトウェアが含まれています。

本製品には、ソフトウェア copyright (C) 2003-2006 Joe Walnes, 2006-2007 XStream Committers が含まれています。All rights reserved. 本ソフトウェアに関する許諾および制限は、<http://j.org/license.html> にある使用条件に従います。本製品には、Indiana University Extreme! Lab によって開発されたソフトウェアが含まれています。詳細については、<http://www.extreme.indiana.edu/> を参照してください。

本製品には、ソフトウェア Copyright (C) 2013 Frank Balluffi and Markus Moeller が含まれています。All rights reserved. 本ソフトウェアに関する許諾および制限は、MIT ライセンスの使用条件に従います。

特許については、<https://www.informatica.com/legal/patents.html> を参照してください。

免責: 本文書は、一切の保証を伴わない「現状渡し」で提供されるものとし、Informatica LLC は他社の権利の非侵害、市場性および特定の目的への適合性の黙示の保証などを含めて、一切の明示的および黙示的保証の責任を負いません。Informatica LLC では、本ソフトウェアまたはドキュメントに誤りのないことを保証していません。本ソフトウェアまたはドキュメントに記載されている情報には、技術的に不正確な記述や誤植が含まれる場合があります。本ソフトウェアまたはドキュメントの情報は、予告なしに変更されることがあります。

## NOTICES

この Informatica 製品 (以下「ソフトウェア」) には、Progress Software Corporation (以下「DataDirect」) の事業子会社である DataDirect Technologies からの特定のドライバ (以下「DataDirect ドライバ」) が含まれています。DataDirect ドライバには、次の用語および条件が適用されます。

1. DataDirect ドライバは、特定物として現存するままの状態を提供され、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任を負わないものとします。国または地域によっては、法律の強行規定により、保証責任の制限が禁じられる場合、強行規定の制限を受けるものとします。
2. DataDirect または第三者は、予見の有無を問わず発生した ODBC ドライバの使用に関するいかなる直接的、間接的、偶発的、特別、あるいは結果的損害に対して責任を負わないものとします。本制限事項は、すべての訴訟原因に適用されます。訴訟原因には、契約違反、保証違反、過失、厳格責任、詐称、その他の不法行為を含みますが、これらに限るものではありません。

発行日: 2023-12-21

# 目次

<b>序文</b> .....	9
<b>第1章: データ型とフィールドのプロパティの理解</b> .....	10
データ型とフィールドプロパティの概要.....	10
データ型とフィールドプロパティ.....	11
日付、時刻、番号の形式設定.....	19
[フィールドのプロパティ] ダイアログの使用.....	20
[フィールドのプロパティ] ダイアログの多くまたは一部のコントロール.....	20
リストを表示: 詳細クエリ.....	20
リストを表示: カスタムリスト.....	21
リストを表示: 子オブジェクトの一覧表示.....	21
リストを表示: なし.....	21
リストを表示: オブジェクトクエリ.....	21
リストを表示: フィールドの選択リスト.....	22
リストを表示: ロール内のユーザー.....	23
式エディタの使用.....	23
関数の使用.....	29
その他の関数.....	38
関数と変数の指定.....	41
デジタル署名関数の概要.....	42
添付ファイル.....	44
HTTP および JMS ヘッダー.....	47
HTTP 動詞の関数.....	50
ヒューマンタスク XQuery ユーティリティ.....	50
単純型のリストを使用したフィールドの作成.....	52
単純型のリストを使用したプロセスの呼び出し.....	53
カスタムタイプを使用したフィールドの作成.....	54
定義された接続タイプを使用したフィールドの作成.....	55
<b>第2章: プロセスのデザイン</b> .....	57
プロセスの作成.....	58
プロセスプロパティの設定.....	58
全般プロパティ.....	59
開始プロパティ.....	60
入力フィールドのプロパティ.....	64
出力フィールドのプロパティ.....	65
一時フィールドのプロパティ.....	66
メッセージのプロパティ.....	67
詳細プロパティ.....	69
付記.....	71

プロセスのフィールド値の更新. . . . .	71
プロセスステップの追加. . . . .	73
割り当てステップ. . . . .	73
サービスステップ. . . . .	74
サブプロセスステップ. . . . .	75
ヒューマンタスクステップ. . . . .	77
作成ステップ. . . . .	77
受信ステップ. . . . .	78
待機ステップ. . . . .	79
マイルストーンステップ. . . . .	80
終了ステップ. . . . .	82
スローステップ. . . . .	83
決定ステップ. . . . .	84
並列パスステップ. . . . .	85
ジャンプステップ. . . . .	86
システムサービスアクション. . . . .	86
オブジェクトの削除. . . . .	86
Data Marketplace API の実行. . . . .	87
JMS キュー追加サービス. . . . .	90
MDM ヒューマンタスク. . . . .	91
シェルコマンドの実行. . . . .	93
クラウドタスクの実行. . . . .	94
フォールト処理. . . . .	95
フォールト処理と境界イベント. . . . .	95
プロセスからフォールトを返す方法. . . . .	97
<b>第 3 章: データの使用と表示. . . . .</b>	<b>99</b>
オブジェクトの選択と表示. . . . .	99
選択リストを使用したフィールドの挿入. . . . .	100
テーブルのフィールドの挿入. . . . .	101
フィールドのテキストとしての入力. . . . .	101
カラムの表示. . . . .	102
ステップでのデータの設定. . . . .	103
ステップでのフィールドの使用. . . . .	104
ソース値の設定. . . . .	105
ソース値の設定: コンテンツ. . . . .	105
ソース値の設定: フィールド. . . . .	106
ソース値の設定: 計算式. . . . .	106
ソース値の設定: クエリ. . . . .	107
ソース値の設定: 画面. . . . .	108
<b>第 4 章: ガイドのデザイン. . . . .</b>	<b>109</b>
ガイドの作成. . . . .	110

ガイドのプロパティの設定. . . . .	111
全般プロパティ. . . . .	112
開始プロパティ. . . . .	112
画面のプロパティ. . . . .	113
入力フィールドのプロパティ. . . . .	118
出力フィールドのプロパティ. . . . .	118
一時フィールドのプロパティ. . . . .	119
結果. . . . .	119
詳細プロパティ. . . . .	121
付記. . . . .	122
ガイドステップの追加. . . . .	123
割り当てステップ. . . . .	123
作成ステップ. . . . .	124
画面ステップ. . . . .	125
決定ステップ. . . . .	127
ジャンプステップ. . . . .	128
プロセス呼び出しステップ. . . . .	128
組み込みガイドステップ. . . . .	128
サービスステップ. . . . .	130
終了/マイルストーンステップ. . . . .	131
ガイドの外観の設定. . . . .	133
ガイドの外観. . . . .	133
ガイドの画面エディタ. . . . .	135
ガイドの検証と保存. . . . .	141
ガイドのシミュレーション. . . . .	142
<b>第 5 章: プロセスオブジェクトのデザイン. . . . .</b>	<b>144</b>
プロセスオブジェクト作成オプション. . . . .	144
プロセスオブジェクトの手動作成. . . . .	145
プロセスでプロセスオブジェクトを使用するためのルールおよびガイドライン. . . . .	147
WSDL ファイルおよび XSD ファイルからのプロセスオブジェクトの作成. . . . .	147
Swagger ファイルからのプロセスオブジェクトの作成. . . . .	155
OpenAPI 3.0 ファイルからのプロセスオブジェクトの作成. . . . .	163
JSON ファイルからのプロセスオブジェクトの作成. . . . .	171
JSON ファイルのルールおよびガイドライン. . . . .	179
必須要素と NULL 可能要素. . . . .	180
プロセスオブジェクトインポートのルールおよびガイドライン. . . . .	182
<b>第 6 章: サービスコネクタのデザイン. . . . .</b>	<b>184</b>
オプション 1: 新しいサービスコネクタの作成. . . . .	186
プロパティの定義. . . . .	186
関数と変数の指定. . . . .	187
アクションの定義. . . . .	189

サービスコネクタプロセスオブジェクトの作成. . . . .	201
サービスコネクタのテスト. . . . .	202
オプション 2: ファイルインポートによるサービスコネクタの生成. . . . .	205
WSDL ファイルからのサービスコネクタの作成. . . . .	205
修飾要素が含まれる WSDL ファイル. . . . .	209
繰り返し要素が含まれる WSDL ファイル. . . . .	210
elementFormDefault. . . . .	211
選択要素. . . . .	211
Swagger JSON ファイルからのサービスコネクタの作成. . . . .	211
Swagger ファイルに関するルールおよびガイドライン. . . . .	214
OpenAPI 3.0 ファイルからのサービスコネクタの作成. . . . .	215
HTTP 操作. . . . .	219
JSON ペイロード. . . . .	221
必須要素と NULL 可能要素. . . . .	222
オプション 3: GitHub からのサービスコネクタのダウンロード. . . . .	224
オプション 4: Informatica Marketplace からのサービスコネクタのダウンロード. . . . .	225
オプション 5: Administrator からのサービスコネクタのダウンロード. . . . .	228
データアクセスサービスコネクタの作成. . . . .	231
プロパティの定義. . . . .	232
変数の指定. . . . .	234
アクションの定義. . . . .	234
プロセスオブジェクトの作成. . . . .	239
データアクセスサービスコネクタのテスト. . . . .	240
サービスコネクタの検証. . . . .	243
サービスコネクタの保存およびパブリッシュ. . . . .	244
<b>第 7 章: コネクタの使用. . . . .</b>	<b>245</b>
アプリケーション接続の作成. . . . .	245
アプリケーション接続プロパティの定義. . . . .	246
イベントソースの指定. . . . .	248
イベントターゲットの指定. . . . .	249
アプリケーション接続の機密フィールド値の更新. . . . .	250
アプリケーション接続の検証. . . . .	252
アプリケーション接続の保存およびパブリッシュ. . . . .	253
パブリッシュ済みメタデータの表示. . . . .	253
アプリケーション接続のパブリッシュ解除. . . . .	254
<b>第 8 章: システムサービス、リスナおよびコネクタ. . . . .</b>	<b>255</b>
odata プロバイダの使用. . . . .	255
サポートされる OData V4 および OData V2 URI 規約. . . . .	256
カスタム複合キー. . . . .	257
Salesforce システムサービスのリスナとコネクタの使用. . . . .	260
Salesforce 送信メッセージ. . . . .	260

Salesforce 接続の使用. . . . .	266
電子メールサービスの使用. . . . .	270
電子メール接続プロパティ. . . . .	270
電子メール接続のパブリッシュ. . . . .	272
プロセスまたはガイドでの電子メールオブジェクトの使用. . . . .	274
プロセスまたはガイドでの電子メール接続の使用. . . . .	275
Java Message Service (JMS) の使用. . . . .	276
シェルコマンド起動サービスコネクタの使用. . . . .	276

## **第 9 章: ヒューマンタスクの設計. . . . . 278**

ヒューマンタスクのユーザーロール. . . . .	279
ガイドレンダリングタスク. . . . .	279
ヒューマンタスクの作成. . . . .	280
手順 1. ヒューマンタスクアセットを定義する. . . . .	281
手順 2. 潜在的な所有者を定義する. . . . .	282
手順 3. 除外する所有者を定義する. . . . .	283
手順 4. タスク管理者の定義. . . . .	283
手順 5. 関係者を定義する. . . . .	284
手順 6. 入力フィールドと出力フィールドを設定する. . . . .	284
手順 7. タスクの結果を設定する. . . . .	286
手順 8. 表示設定を指定する. . . . .	286
プロセス内でのヒューマンタスクの使用. . . . .	287



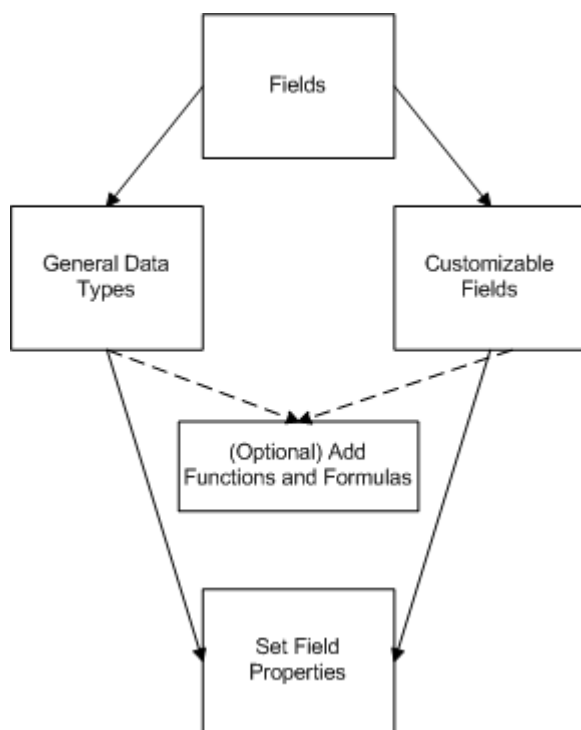
# 序文

『デザイン』を使用して、プロセス、ガイド、プロセスオブジェクト、サービスコネクタ、システムサービス、リスナ、コネクタなどのアセットの作成方法を確認します。

# 第 1 章

## データ型とフィールドのプロパティの理解

次の図は、この章で学ぶ内容を示しています。

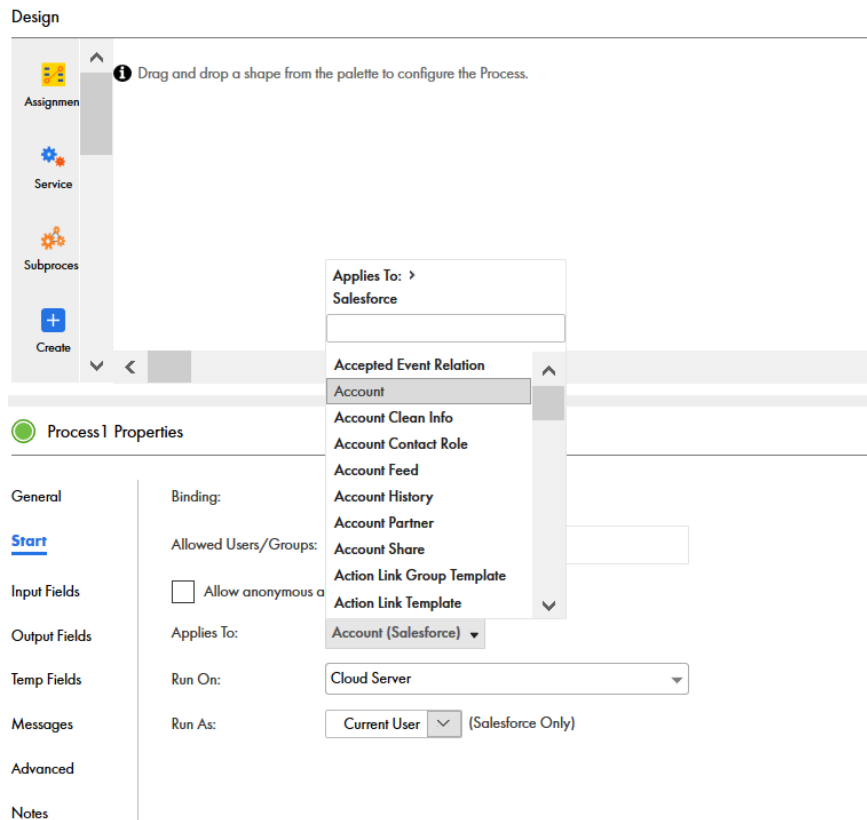


## データ型とフィールドプロパティの概要

Process Designer は、アプリケーションデータベース内に格納された情報を使用します。各データベーステーブルに、プロセス内で必要なフィールドを操作するために、接続を介してアクセスできるオブジェクトタイプが格納されます。

プロセス、ステップのプロパティを定義するときに、オブジェクトを表示し、関連付けられたフィールドをリストから選択するために、ドリルダウンします。

例えば、次の場合、Salesforce オブジェクトによって、プロセスで使用できる多数のオブジェクトタイプにアクセスできます。



多くの場合、プロセスによって、アプリケーションに基づいて格納されるデータ型が決まります。その他のフィールドの場合、それらを単純型として定義し、該当するデータ型を選択する必要があることがあります。データ型とは別に、その他のフィールドプロパティを制御する必要がある場合があります。例えば、デフォルト値や、ユーザーを補助するためのホバーテキストを指定できます。

フィールドプロパティを更新するには、編集するフィールドをクリックします。

フィールドプロパティとデータ型の詳細については、このセクションのトピックを参照してください。

## データ型とフィールドプロパティ

Informatica Process Designer で、API データ型とオプションから派生したフィールドプロパティを指定できます。フィールドプロパティは、次のいずれかにできます。

- 一般的なデータ型プロパティ。例えば、[プロセスのプロパティ] の【フィールド】タブで指定したものです。
- コンテキストに合わせて調整できるフィールド。例えば、画面ステップの選択リストです。

[フィールドのプロパティ] 内の多くのデータ型では、次の基本オプションを使用できます。

- **デフォルト値**または**初期値**: 項目が最初に表示されるとき値。
- **フィールド**: 【フィールドのプロパティ】ダイアログの呼び出し元フィールドの名前。
- **ホバーテキスト**: ユーザーがフィールド上にカーソルを置いたときに表示されるテキスト。
- **必須**: 選択すると、このフィールドの値を指定する必要があります。

- **リストを表示:** 表示される値は、ここで設定される値に基づいて実行時に決定されます。

次の表で、組み込みのデータ型について説明します。

**注:** [フィールドのプロパティ] にアクセスする場所によって、使用可能なデータ型のオプションは異なる場合があります。

データ型	説明
任意	<p>[参照] オプションを設定せずにプロセスオブジェクトのフィールドを宣言する場合、任意データ型を使用します。つまり、データ型はオブジェクトですが、どのオブジェクトであるかを指定する必要はありません。任意オブジェクトを使用する場合、オブジェクトにさらにドリルダウンすることはできません。</p> <p>一部のサービスでは大量のデータが返されますが、必要なデータはごく一部である場合があります。任意データ型を使用して、返されるすべてのデータを限られたフィールドにモデル化します。</p> <p>例えば、REST プロキシを使用すると、受信したペイロードを入力として渡して、このペイロードに回答します。REST プロキシは要求または回答に対して何らかの処理を行い、メッセージ全体を理解する必要はありません。</p> <p>2番目の例は、異種リストをサポートする例です。単純な REST API では異種リストが含まれない傾向がありますが、SOAP ベースの API には含まれます。例えば、SOAP API には、クエリ自体に依存するデータを返すクエリ操作が含まれる場合があります。ここでは、クエリ操作の出力の static 型は、任意型のオブジェクトリストになります。</p> <p>3番目の例は、クエリに一致するレコードを返すサービスを呼び出す例です。各レコードに一意的なフィールドのセットがあり、そのため、設計時にそれらを知ることはありません。実行時に、各フィールドは、フィールド名に対応するタグ名を使用して、レコード要素の子要素に変換されます。変換された JSON の戻り値は、次のように出力されます。</p> <pre>{ "record": [ { "ROWID_OBJECT": "2001", "CREATOR": "admin", ... }, { "ROWID_OBJECT": "2002", "CREATOR": "admin", ... }, ... ] }</pre>
添付、添付ファイル	<p>添付および添付ファイルのプロセスを使用して添付ファイルを渡し、応答の関数を使用してファイルサイズやファイル名などのデータを抽出します。</p> <p>デフォルトの最大ファイルサイズは 5,242,880 バイトであり、デフォルトの最大ファイル数は 10 です。これらのフィールドプロパティは変更できます。</p> <p><b>注:</b> プロセスを Cloud Server 上で実行する場合、サイズがデフォルトの 5,242,880 バイトを超える添付ファイルは使用しないでください。クラウドサーバーは、5,242,880 バイトを超える添付ファイルを処理できません。</p>
チェックボックス	<p>チェックボックス型を使用すると、ユーザーは真偽判定を行うことができます。この決定は、[はい] / [いいえ] または [いいえ] / [はい] として表示することもできます。例えば、チェックボックスを使用して、住所が職場の電話番号か自宅の電話番号かを示すことができます。</p> <p>代わりに、チェックボックスを [はい] / [いいえ] の値として表示することもできます。</p> <p>- <b>次として表示:</b> チェックボックス、[はい] / [いいえ]、および [いいえ] / [はい] の項目を含む選択リスト。この選択肢によって、ユーザーに表示されるラベルが設定されます。</p> <p>チェックボックスデータ型のリストは、プロセス内の入力フィールド、出力フィールド、一時フィールド、およびメッセージにも設定できます。Boolean 値のカンマ区切りリストを入力できます。</p> <p>チェックボックスのリストをデータ型として設定するには、データ型のフィールドとして <b>[単純型] &gt; [チェックボックス]</b> を選択し、次に <b>[タイプの編集]</b> ダイアログボックスで <b>[このタイプのオブジェクトのリストを許可します]</b> チェックボックスを選択します。</p> <p>出力フィールドのチェックボックスデータ型のリストに初期値を割り当てる場合は、次の形式で値を入力できます。</p> <p>(true())または(false())</p> <p>単純型のリストの使用方法に関する詳細については、<a href="#">「単純型のリストを使用したフィールドの作成」</a> (ページ 52)を参照してください。</p>

データ型	説明
通貨	<p>通貨フィールドには通貨での値が格納されます。数値や小数点に加えて、必要に応じてカンマを使用することもできます。カンマは常に省略可能です。</p> <ul style="list-style-type: none"> <li>- <b>長さ/小数点以下の桁数:</b> 入力可能な数値の数を指定します (左のボックス)。右のボックスを使用して、小数点以下の桁数の最大値を入力します。0を入力すると、小数点の右側に桁はありません。</li> <li>- <b>通貨記号を表示しない:</b> 選択すると、通貨記号は表示されません。</li> </ul>
日付	<p>出力される日付は UTC であり、出力される日時は ISO 8601 形式です。例えば、入力日付を 2016-03-29 と入力し、currentTime XQuery 関数を使用して日付と時刻の値を取得すると、次の出力が表示されます。</p> <ul style="list-style-type: none"> <li>- 日付: 2016-03-29Z</li> <li>- dateTime: 2016-03-29T06:00:48.525Z</li> <li>- 時刻: 06:00:48.525Z</li> </ul> <p>有効な日付値のサンプルを次に示します。</p> <ul style="list-style-type: none"> <li>- 2001-10-26</li> <li>- 2001-10-26+02:00</li> <li>- 2001-10-26Z</li> <li>- 2001-10-26+00:00</li> <li>- -2001-10-26</li> <li>- -2000-04-01</li> </ul> <p>タイムゾーンを渡さない場合、Informatica Process Designer は時間が UTC であると見なします。エンドポイント URL で日付をクエリパラメータとして渡す場合に、その日付に特殊文字が含まれているときは、日付をエンコードする必要があります。例えば、2001-10-26+02:00 を 2001-10-26%2B02%3A00 にエンコードします。</p> <p>日付が表示される形式を指定する必要がある場合、「日付、時刻、番号の形式設定」を参照してください。</p> <p>入力フィールド用の日付データ型のリストは、プロセス内の入力フィールド、出力フィールド、一時フィールド、およびメッセージにも設定できます。日付値のカンマ区切りリストを入力できます。</p> <p>日付のリストをデータ型として設定するには、データ型のフィールドとして <b>[単純型] &gt; [日付]</b> を選択し、次に <b>[タイプの編集]</b> ダイアログボックスで <b>[このタイプのオブジェクトのリストを許可します]</b> チェックボックスを選択します。</p> <p>単純型のリストの使用方法に関する詳細については、<a href="#">「単純型のリストを使用したフィールドの作成」 (ページ 52)</a> を参照してください。</p>

データ型	説明
日付/時刻	<p>[日付/時刻] フィールドをプロセスで選択すると、ユーザーは時刻を一覧から選択できます。</p> <ul style="list-style-type: none"> <li>- <b>デフォルト値:</b> 特定の日付、フィールド、または間隔を入力します。</li> <li>- <b>30分増分:</b> 選択した場合、時刻選択リストに正時と30分単位の時間が表示されます。選択しない場合、ユーザーは任意の時刻の値 (10:37 など) を入力できます。</li> </ul> <p>出力される時刻は UTC であり、出力される日時は ISO 8601 形式です。例えば、入力日付を 2016-03-29 と入力し、currentTime XQuery 関数を使用して日付と時刻の値を取得すると、次の出力が表示されます。</p> <ul style="list-style-type: none"> <li>- 日付: 2016-03-29Z</li> <li>- dateTime: 2016-03-29T06:00:48.525Z</li> <li>- 時刻: 06:00:48.525Z</li> </ul> <p>有効な dateTime 値のサンプルを次に示します。</p> <ul style="list-style-type: none"> <li>- 2001-10-26T21:32:52</li> <li>- 2001-10-26T21:32:52+02:00</li> <li>- 2001-10-26T19:32:52Z</li> <li>- 2001-10-26T19:32:52+00:00</li> <li>- -2001-10-26T21:32:52</li> <li>- 2001-10-26T21:32:52.12679</li> </ul> <p>タイムゾーンを渡さない場合、Informatica Process Designer は時間が UTC であると見なします。エンドポイント URL で日付と時刻をクエリパラメータとして渡す場合に、その日付と時刻に特殊文字が含まれているときは、日付をエンコードする必要があります。例えば、2001-10-26T19:32:52+00:00 を 2001-10-26T19%3A32%3A52%2B00%3A00 にエンコードします。</p> <p>日時が表示される形式を指定する必要がある場合、「日付、時刻、番号の形式設定」を参照してください。</p> <p>入力フィールド用の日付/時刻データ型のリストは、プロセス内の入力フィールド、出力フィールド、一時フィールド、およびメッセージにも設定できます。日付と時刻の値のカンマ区切りリストを入力できます。</p> <p>日付と時刻のリストをデータ型として設定するには、データ型のフィールドとして <b>[単純型] &gt; [日付/時刻]</b> を選択し、次に <b>[タイプの編集]</b> ダイアログボックスで <b>[このタイプのオブジェクトのリストを許可します]</b> チェックボックスを選択します。</p> <p>単純型のリストの使用方法に関する詳細については、<a href="#">「単純型のリストを使用したフィールドの作成」 (ページ 52)</a> を参照してください。</p>
電子メール	<p>このフィールドのコンテンツは電子メールアドレスです。</p> <ul style="list-style-type: none"> <li>- <b>表示文字#:</b> このアドレス用にプロセス内に表示する文字数。</li> <li>- <b>最大文字#:</b> このアドレス用に入力できる最大文字数。</li> </ul>
フォーマットされたテキスト	<p>入力テキスト内にユーザーが入力できる文字の種類を表す形式。</p> <ul style="list-style-type: none"> <li>- <b>形式:</b> ユーザーが入力するテキストのパターン。形式を定義する場合、以下の文字を使用できます。 <ul style="list-style-type: none"> <li>- a: a~z までの小文字。</li> <li>- 9: 数値。</li> <li>- *: 任意の文字、数値、記号。</li> </ul> </li> </ul> <p>注: 「-」はリストに表示されません。この文字を入力できる場合、ユーザーにステップが表示されるときに、その文字が表示されます。</p>
イメージ	<p>画像を格納するフィールド。最もよく使用されるのは、自動アクションによって画像が取得されるときです。</p> <ul style="list-style-type: none"> <li>- <b>プレビューの表示:</b> 画像のプレビューが表示されます。</li> <li>- <b>最大幅:</b> 画像が表示される領域の最大幅。</li> <li>- <b>最大高:</b> 画像が表示される領域の最大高。</li> <li>- <b>表示文字#:</b> このアドレス用にプロセス内に表示する文字数。</li> </ul>

データ型	説明
整数	<p>正または負の整数。</p> <p>ユーザーが入力できる数値の最大値を入力するには、<b>【数字】</b> フィールドを使用します。</p> <p>数値はユーザーのロケールの形式を使用して表示されます。</p> <p>数値が表示される形式を指定する必要がある場合、「日付、時刻、番号の形式設定」を参照してください。</p> <p>入力フィールド用の整数データ型のリストは、プロセス内の入力フィールド、出力フィールド、一時フィールド、およびメッセージにも設定できます。整数値のカンマ区切りリストを入力できます。</p> <p>整数のリストをデータ型として設定するには、データ型のフィールドとして <b>【単純型】</b> &gt; <b>【整数型】</b> を選択し、次に <b>【タイプの編集】</b> ダイアログボックスで <b>【このタイプのオブジェクトのリストを許可します】</b> チェックボックスを選択します。</p> <p>単純型のリストの使用方法に関する詳細については、<a href="#">「単純型のリストを使用したフィールドの作成」 (ページ 52)</a> を参照してください。</p>
複数選択リスト	<p>ユーザーが 1 つまたは複数の行を選択できる選択リストフィールド。複数の項目を選択するには、ユーザーは Ctrl キーを押しながら各項目をクリックします。</p> <ul style="list-style-type: none"> <li>- <b>高さ (行数)</b> : プロセス内に表示する 1 つのリスト項目がそれぞれに格納される行数。</li> <li>- <b>カンマ区切りリスト</b> : リスト内の項目のリスト。リスト内の各項目は、カンマで区切ります。指定する各項目は、実行時にリスト内で使用できます。</li> </ul> <p><b>使用上の注意</b></p> <ul style="list-style-type: none"> <li>- 入力フィールドのデータ型を複数選択リストまたは選択リストとして宣言し、設定値を入力しない場合、カンマの代わりに、区切り文字としてセミコロン (;) を使用する必要があります。<b>【次の値に等しい】</b> 条件にはセミコロンを使用し、<b>【次を含む】</b> 条件にはカンマを使用することに注意してください。</li> <li>- <b>【複数選択リスト】</b> プロパティまたは <b>【選択リスト】</b> プロパティで使用可能な値が指定されておらず、検索サービスから値のリストを受け取った場合、次の操作を行う必要があります。 <ol style="list-style-type: none"> <li>1. 検索サービスを指定します。さらに、</li> <li>2. 他のステップ内で画面ステップまたは入力オプションからリストを入力します。利用可能な値のリストは、現在のステップ内でのみ存在することに注意してください。他のステップで同じリストを使用するには、検索サービスを再度指定します。</li> </ol> </li> <li>- また、割り当てステップ内で未定義の複数選択リストまたは選択リストに値を割り当てることもできます。その場合、<b>【ソース】</b> として <b>【計算式】</b> を使用し、選択リストまたは複数選択リストの値をコンテンツとして使用します。数式を入力するときは、値を引用符で囲みます。例えば、選択リストの場合 <code>"value"</code>、複数選択リストの場合 <code>"value1;value2"</code> です。</li> </ul>
番号	<p>正または負の 10 進数。</p> <p>入力可能な数値の数 (左のボックス) を指定するには <b>【長さ/小数点以下の桁数】</b> フィールドを使用します。右のボックスを使用して、小数点以下の桁数の最大値を入力します。0 を入力すると、小数点の右側に桁はありません。</p> <p>数値はユーザーのロケールの形式を使用して表示されます。</p> <p>数値が表示される形式を指定する必要がある場合、「日付、時刻、番号の形式設定」を参照してください。</p> <p>入力フィールド用の数値データ型のリストは、プロセス内の入力フィールド、出力フィールド、一時フィールド、およびメッセージにも設定できます。数値のカンマ区切りリストを入力できます。</p> <p>数値のリストをデータ型として設定するには、データ型のフィールドとして <b>【単純型】</b> &gt; <b>【数値型】</b> を選択し、次に <b>【タイプの編集】</b> ダイアログボックスで <b>【このタイプのオブジェクトのリストを許可します】</b> チェックボックスを選択します。</p> <p>単純型のリストの使用方法に関する詳細については、<a href="#">「単純型のリストを使用したフィールドの作成」 (ページ 52)</a> を参照してください。</p>

データ型	説明
オブジェクト ID	<p>オブジェクト ID は、オブジェクトインスタンスの ID です。例えば、リードの 1 つを識別します。(この値は、アプリケーションによってオブジェクトに割り当てられます)。</p> <p>このダイアログ固有の部分については、この表の後に説明します。。ここと [フィールドのプロパティ] ダイアログ内で使用されるフィールドについては、このトピックの最初で説明しています。</p>
オブジェクトリスト	<p>オブジェクトリストはオブジェクト ID のセットです。オブジェクト ID は、オブジェクトインスタンスの ID です。例えば、リードの 1 つを識別します。</p> <ul style="list-style-type: none"> <li>- <b>参照:</b> リストに含まれるオブジェクトのタイプ。オブジェクトのラベルでなく、オブジェクトタイプの正式名称を入力する必要があります。リストに複数のオブジェクトタイプを含められる場合、カンマ区切りのオブジェクトタイプのリストを入力できます。ただし、1 つの参照タイプを持つのでない限りフィールドに提供できないクエリとテーブルの機能があるため、これは避ける必要があります。</li> <li>- <b>表示フィールド:</b> オブジェクト ID に関連付けられたデータからのカラム (手動で入力する必要があります)。</li> <li>- <b>高さ (行数):</b> プロセス内に表示する 1 つのリスト項目がそれぞれに格納される行数。</li> <li>- <b>幅 (%) :</b> 情報の表示に使用される領域の割合。</li> </ul> <p>関連付けられたクエリが、より多くの行を返すように要求する LIMIT 句を使用している場合、読み取り専用のオブジェクトリストには 100 行の制限があります。</p> <p>その他のフィールドについては、このトピックの最初で説明しています。</p>
割合	<p>割合で表される数値。</p> <ul style="list-style-type: none"> <li>- <b>精度:</b> 数値の最大桁数。</li> <li>- <b>スケール:</b> 小数点以下の桁数の最大値。0 を入力すると、小数点の右側に桁はありません。</li> </ul> <p>数値はユーザーのロケールの形式を使用して表示されます。</p> <p>ここにいくつかの例を示します。これらの例では、次のようにそれぞれ 100% を超える数値を示しています。</p> <ul style="list-style-type: none"> <li>- 精度 5、スケール 2: 123.45 または 123.4 または 123. または 123</li> <li>- 精度 5、スケール 0: 12345 または 12 または 12,345</li> </ul> <p>割合が表示される形式を指定する必要がある場合、「日付、時刻、番号の形式設定」を参照してください。</p>
電話番号	<p>電話番号を入力できるフィールド。</p> <ul style="list-style-type: none"> <li>- <b>形式:</b> 数値を表す文字 (9 とその他の表示文字) を入力します。例えば、99-999 はプロセス内で「__-__」と表示されます。</li> </ul>
選択リスト	<p>ユーザーが 1 行を選択できる選択リストフィールド。</p> <ul style="list-style-type: none"> <li>- <b>カンマ区切りリスト:</b> リスト内の項目のリスト。リスト内の各項目は、カンマで区切ります。プロセス内で、各項目がそれぞれの行に表示されます。</li> </ul> <p><b>使用上の注意</b></p> <ul style="list-style-type: none"> <li>- テキストをユーザーに表示し、フィールドに別の値を割り当てる場合は、等号 (=) を使用してラベルを値から区切ります。例えば、「赤,白,青」を値として入力すると、ユーザーにこれらの 3 つの文字列が表示され、選択した文字列がフィールドに割り当てられます。これとは異なり、「赤=1,白=2,青=3」と入力すると、ユーザーには同じ 3 つの文字列が表示されますが、選択した項目の数値がフィールドに割り当てられます。</li> <li>- 検索サービスから値のリストを受け取る方法については、複数選択リストの「使用上の注意」も参照してください。</li> </ul>
リッチテキスト領域	<p>HTML コマンドとテキストが格納されるフィールド。HTML エディタを使用してこのテキストを入力します。</p> <ul style="list-style-type: none"> <li>- <b>高さ (行数):</b> ユーザーがテキストを入力できる、プロセスが表示する領域内の行数。</li> <li>- <b>幅 (%) :</b> プロセスの幅に対応するテキスト領域の幅。</li> </ul> <p>リッチテキスト領域に JavaScript を入力することはできません。</p>



データ型	説明
テキスト	<p>ユーザーがテキストを入力できる文字のサイズと数を変更するコントロール。</p> <ul style="list-style-type: none"> <li>- <b>最大文字#</b>: テキストの入力時に入力できる最大文字数。</li> <li>- <b>表示文字#</b>: プロセス内に表示する文字数。プロセスのサイズは制限されるため、このフィールドを使用して、ユーザーの入力内容を表示する文字数を指定します。</li> </ul> <p>入力フィールド用のテキストデータ型のリストは、プロセス内の入力フィールド、出力フィールド、一時フィールド、およびメッセージにも設定できます。テキスト値のカンマ区切りリストを入力できます。</p> <p>テキストのリストをデータ型として設定するには、データ型のフィールドとして <b>[単純型]</b> &gt; <b>[テキスト]</b> を選択し、次に <b>[タイプの編集]</b> ダイアログボックスで <b>[このタイプのオブジェクトのリストを許可します]</b> チェックボックスを選択します。</p> <p>単純型のリストの使用方法に関する詳細については、<a href="#">「単純型のリストを使用したフィールドの作成」 (ページ 52)</a> を参照してください。</p>
テキスト領域	<p>ユーザーがテキストを入力できるコントロール。これは、1 行または複数の行として表示できます。</p> <ul style="list-style-type: none"> <li>- <b>幅 (%)</b>: プロセスの幅に対応するテキスト領域の幅。</li> <li>- <b>高さ (行数)</b>: ユーザーがテキストを入力できる、プロセスが表示する領域内の行数。</li> </ul>
時間	<p>選択リストから時刻を選択するには、<b>[時間]</b> フィールドを選択します。</p> <p>表示されるフィールドのプロパティダイアログは次のとおりです。</p> <p>出力される時刻は UTC であり、出力されるデータと時刻は ISO 8601 形式です。例えば、入力日付を 2016-03-29 と入力し、currentTime XQuery 関数を使用して日付と時刻の値を取得すると、次の出力が表示されます。</p> <ul style="list-style-type: none"> <li>- 日付: 2016-03-29Z</li> <li>- dateTime: 2016-03-29T06:00:48.525Z</li> <li>- 時刻: 06:00:48.525Z</li> </ul> <p>有効な時間値のサンプルを次に示します。</p> <ul style="list-style-type: none"> <li>- 21:32:52</li> <li>- 21:32:52+02:00</li> <li>- 19:32:52Z</li> <li>- 19:32:52+00:00</li> <li>- 21:32:52.12679</li> </ul> <p>タイムゾーンを渡さない場合、Informatica Process Designer は時間が UTC であると見なします。<b>[30 分増分]</b> を選択すると、時刻選択リストに正時と 30 分のリストが表示されます。選択しないと、任意の時間の値を入力できます。例えば、「10:37」のように入力できます。</p> <p>エンドポイント URL で時間をクエリパラメータとして渡す場合に、その時間に特殊文字が含まれているときは、時間をエンコードする必要があります。例えば、21:32:52+02:00 を 21%3A32%3A52%2B02%3A00 にエンコードします。</p> <p>時刻が表示される形式を指定する場合、「日付、時刻、番号の形式設定」を参照してください。</p> <p>入力フィールド用の時間データ型のリストは、プロセス内の入力フィールド、出力フィールド、一時フィールド、およびメッセージにも設定できます。時間値のカンマ区切りリストを入力できます。</p> <p>時間のリストをデータ型として設定するには、データ型のフィールドとして <b>[単純型]</b> &gt; <b>[時間]</b> を選択し、次に <b>[タイプの編集]</b> ダイアログボックスで <b>[このタイプのオブジェクトのリストを許可します]</b> チェックボックスを選択します。</p> <p>単純型のリストの使用方法に関する詳細については、<a href="#">「単純型のリストを使用したフィールドの作成」 (ページ 52)</a> を参照してください。</p>

データ型	説明
URL	<p>URL を格納するフィールド。</p> <ul style="list-style-type: none"> <li>- <b>読み取り専用表示用:</b> 選択肢は <code>[リンク]</code>、<code>[ボタン]</code>、または <code>[IFrame]</code> です。</li> <li>- <b>読み取り専用ラベル用:</b> フィールドのコンテンツを特定するテキスト。</li> <li>- <b>高さ:</b> URL が表示される領域の高さ。</li> <li>- <b>幅:</b> URL が表示される領域の幅。</li> <li>- <b>表示文字#:</b> プロセス内に表示する文字数。プロセスのサイズは制限されるため、このフィールドを使用して、ユーザーの入力内容を表示する文字数を指定します。</li> </ul> <p>URL は、入力、出力、または一時プロセスフィールドであるか、それに基づくフィールドにできます。フィールドのデータは、有効な URL であると見なされます。</p> <p>画面ステップ内でボタンを作成する手順</p> <ol style="list-style-type: none"> <li>1. <code>[フィールド]</code> タブをプロセスプロパティ内で使用して、フィールドを作成します。</li> <li>2. 割り当てステップを使用して、リンク先の URL にデータを割り当てます。コンテンツになるこのデータには、<code>{!FieldName}</code> という構文を使用して、ID フィールドなどの他のフィールドへの参照を含めることができます。Process Designer は、テキスト入力領域の右側のボタンを使用してデータを自動的に挿入します。これは、他のページや他のデータに依存するハイパーリンクにリンクするときに非常に便利です。</li> <li>3. 割り当てステップの後の画面で、読み取り専用フィールド選択リストからファイルを選択して、フィールドをステップ内に配置します。</li> <li>4. フィールドの <code>[...]</code> ボタンを選択し、プロパティを指定します。<code>[読み取り専用表示用]</code> を <code>[ボタン]</code> に変更し、<code>[読み取り専用ラベル用]</code> のボタンに表示するテキストを入力します。</li> </ol>
XML	<p>XML データ型は、プロセスおよびサービスコネクタで使用できます。</p> <p>プロセスで XML データ型を使用するには、次のいずれかの方法を選択します。</p> <ul style="list-style-type: none"> <li>- XML データ型をプロセスの割り当てステップで使用する。</li> <li>- XML データ型を JMS または AMQP で使用する。この場合は、ペイロードを XML データにすることができ、メッセージを送受信するプロセスを設計できます。</li> </ul> <p>JMS または AMQP で XML データを使用する場合は、ペイロードをルート要素内にラップします。</p> <p>単純な XML データ要素を他のフィールドにマッピングするには、計算式と XQuery 関数を使用します。</p> <p>例えば、次の XML を考えます。</p> <pre>&lt;o:order xmlns:o="urn:purchasing:system"&gt;   &lt;o:number&gt;123&lt;/o:number&gt;   &lt;o:amount&gt;9213.32&lt;/o:amount&gt; &lt;/o:order&gt;</pre> <p>この XQuery 関数を使用して、注文番号を検索できます。</p> <pre>\$input.myInboundOrder/*:number/text()</pre> <p>名前空間で XML データをクエリするときは、アスタリスク (*) 文字を使用する必要があります。サービスコネクタで XML データ型を使用するには、<code>[アクション]</code> タブの入力フィールドと出力フィールドでタイプに <code>[XML]</code> を選択します。</p>

## オブジェクト ID

オブジェクト ID は、オブジェクトインスタンスの ID です。例えば、リードの 1 つを識別します。(この値は、アプリケーションによってオブジェクトに割り当てられます)。

- **参照:** フィールドの参照先オブジェクトのタイプを示します。オブジェクトのラベルでなく、正式名称を入力する必要があります。オブジェクトが複数のタイプを参照できる場合、カンマ区切りのオブジェクトタイプのリストを入力できます。ただし、1つの参照タイプを持つのでない限りフィールドに提供できないクエリとテーブルの機能があるため、これは避ける必要があります。

多形性関係は、参照されるオブジェクトが、いくつかの異なるオブジェクトタイプのうちの1つである関係です。オブジェクト ID が複数のオブジェクトを参照できる場合は、多形性フィールドを選択します。これらのオブジェクトを選択するための選択リストが使用できるようになります。例えば、アカウントオブジェクトが選択されている間、参照される可能性があるオブジェクトは「受け入れられたイベントの関係」、「アカウントの連絡先」などです。このリストの内容

- **表示フィールド:** オブジェクト ID に関連付けられたデータからのカラム（手動で入力する必要があります）。

他のステップでこのフィールドを使用すると、表示される情報は異なります。Process Designer の例を挙げます。フィールドを挿入するときは、2つの選択肢があります。

最初のフィールド (PolymorphicID) を挿入し、挿入したフィールドの [...] をクリックすると、表示される【フィールドのプロパティ】ダイアログに、[プロパティ] ダイアログで選択されたすべての項目が表示されます。

ただし、[更新対象フィールドを挿入する] 選択リストの [多形性] セクション内から項目を選択するときに表示されるのは、名前が付けられた個々の項目です。実際に必要な項目に戻るには、もう一度クリックする必要があります。

## オブジェクト ID の付記とコメント

オブジェクト ID がキャンバス内に読み取り専用フィールドとして配置されるとき、またはテーブル内のカラムに配置されるときに、オブジェクト ID はオブジェクトへのリンクとして表示されます。オブジェクト ID を数値として表示するには、テキストフィールドに「{!Id}」と挿入します。ただし、ID フィールドが [ルックアップ] ダイアログからの結果に含まれている場合、ID の値が表示されます。

オブジェクト ID がカラムまたは選択リストに表示されるときに、Process Designer では意味がある名前が表示されます。例えば、b6f0f0b2-9f38-4771-8365-58eb4f7b7d41 は「Acme」として表示されます。ただし、オブジェクト ID フィールドが [ルックアップ] ダイアログの結果に含まれている場合は、ID の値を使用して表示されます。例えば、b6f0f0b2-9f38-4771-8365-58eb4f7b7d41 と表示されます。

コンテンツフィールドを使用してオブジェクト ID を入力し、サンドボックスを使用する場合は、プロセスを本番環境に移動するときに、ID を調整する必要があります。コンテンツを使用するだけでなく、クエリを使用して、オブジェクトの名前でオブジェクト ID を読み込みます。

## 日付、時刻、番号の形式設定

形式の値は、【フィールドのプロパティ】ダイアログ ([「データ型とフィールドプロパティ」](#) (ページ 11) を参照) 内、または [ソース] を [計算式] に設定したフィールドから起動する関数内で設定出来ます。

形式の出力は表示値であり、したがって文字列です。

infa:format()関数には、形式を指定するために使用出来る4つの引数があります。例えば、[ソース] を [計算式] に設定した場合、次のような式を使用します。

```
infa.format($output.Created Lead.AnnualRevenue * .75, "#####.")
```

Process Designer は、形式文字列のテキストを評価し、値をその形式で配置します。例えば、「#,##0.00」という形式の場合、Process Designer に、小数点の左側に1桁以上の数値、右側に2桁の数値が表示される必要があることが伝わります。それらが無い場合、ゼロが使用されます。また、長さによって、カンマが挿入される場合があります。

詳細については、「[関数の使用](#)」 (ページ 29)を参照して下さい。

値の形式設定に使用出来るパターンは、Java クラスに使用されるのと同じパターンです。必要に応じて、次の形式についての情報を参照して下さい。

- 日付については、<http://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html> (SimpleDateFormat Java クラス) を参照して下さい。
- 数値については、<http://docs.oracle.com/javase/7/docs/api/java/text/DecimalFormat.html> (DateFormat クラス) を参照して下さい。

## [フィールドのプロパティ] ダイアログの使用

プロセス内でステップを定義するとき、サービスから受け取る、またはプロセス内の別のステップに渡すために必要な入力、出力、その他の変数を処理するために、フィールドを挿入します。追加する各フィールドに、フィールドソース、形式、表示プロパティ、リスト処理を決定するために設定できるフィールドプロパティのセットがあります。

一部のフィールドの場合、式エディタを使用して、フィールドの値とその他の属性を決定するための複合計算式を定義することもできます。

この項のトピックに、[フィールドのプロパティ] ダイアログ内のオプションと、フィールドプロパティと一緒に使用するために記述できるクエリの種類について説明します。

## [フィールドのプロパティ] ダイアログの多くまたは一部のコントロール

ほとんどの [フィールドのプロパティ] ダイアログには、次のコントロールがあります。

- **デフォルト値:** 表示される情報のデフォルト値を設定します。この選択肢には [コンテンツ] (オブジェクトタイプ)、[フィールド] (そのオブジェクトタイプを格納)、[計算式]、[画面] (ユーザー指定のオブジェクト) があります。
- **表示フィールド:** カラム内に表示されるフィールドに名前を付け、表示順を指定します。
- **フィールド:** [フィールドのプロパティ] ダイアログの起動元フィールドの名前を指定します。
- **高さ (行):** 表示される行数を設定します。
- **ホバーテキスト:** ユーザーがフィールドの上にカーソルを置いたときに表示されるテキストを示します。
- **参照:** フィールドの適用先オブジェクトのタイプを示します。この値は、すべてのダイアログボックスで変更できるわけではありません。
- **必須:** この値の入力が必須であるか、削除して置き換えられないようにする場合、このチェックボックスを選択します。
- **リストを表示:** オブジェクトの配置方法を定義します。
- **幅 (%):** 表示される情報の幅をプロセスの幅の割合として設定します。

## リストを表示: 詳細クエリ

[**詳細クエリ**] を選択すると、リストのために取得する情報を詳細に調整できます。

クエリの実行後、デフォルトのアクションは、値がオブジェクト ID であるリストを返すことです。このリストには、表示するための名前フィールドもあります。

クエリの4つのパラメータのソースは、コンテンツ、フィールド、計算式、画面です。ほとんどの場合、ソースはコンテンツです。設定できる項目は次のとおりです。

- **オブジェクトタイプ:** クエリが実行されるオブジェクトタイプ。
- **WHERE 句:** SQL WHERE 句を入力します。ここに入力するテキストには、WHERE キーワードは含まれません。ここで、コンテンツは WHERE 句を記述することを意味し、フィールドはそのフィールドに WHERE 句が含まれることを意味します。
- **表示フィールド:** ここに値を設定すると、現在のフィールド以外のフィールドが返されます。Process Designer がこのデータを取得し、それを選択リスト内に表示します。フィールドを指定しないと、Process Designer は *namefield* (複数のフィールドがある場合は最初のフィールド) を使用します。
- **値フィールド:** Process Designer がデフォルトで使用する値とは異なる値を返します。値フィールドを設定しないと、返されるオブジェクトの ID が使用されます。

**詳細方法:** 詳細クエリを使用して、ステップ内でクエリの結果を表示するステップを作成できます。これは、別個の「テスト」目的のガイドで使用できます。

## リストを表示: カスタムリスト

**[カスタムリスト]** を選択すると、独自の値リストを入力できます。例えば、ニューイングランドの州名を個別に入力できます。プロセスが表示されるときに、この選択リスト内に入力する項目がユーザーに表示されます。

複数の項目はカンマ (,) を使用して区切る必要があります。

デフォルト値を設定することもできます。

## リストを表示: 子オブジェクトの一覧表示

クエリに関連する (または子) オブジェクトを使用します。**[子オブジェクトの一覧表示]** の使用は、フィールド内の関連オブジェクトの使用と類似していますが、これを使用する方が、クエリの条件を指定する必要がないため簡単です。

**[子オブジェクトの一覧表示]** の選択後、子のオブジェクトのタイプも指定する必要があります。ソースの選択肢は、自動的に **[コンテンツ]** に設定され、許可されているソースはこれのみです。

## リストを表示: なし

ユーザーがフィールドに任意の値を指定できる検索ボックスを表示するには、この設定を使用します。フィールドがオブジェクトへの参照である場合、**[リストを表示]** を使用せず、ユーザーが **[参照]** 領域に指定されたタイプのオブジェクトを検索できるようにします。

**[カラムの表示]** 項目を使用して、オブジェクトに関する情報を表示する方法を制御します。

## リストを表示: オブジェクトクエリ

クエリを使用して、Where 句で選択されたすべてのオブジェクトからオブジェクト情報を取得します。

**[オブジェクトクエリ]** を選択すると、Process Designer はオブジェクトクエリコントロールを **[フィールドのプロパティ]** ダイアログに追加します。**[WHERE 句]** の **[ソース]** は常に **[コンテンツ]** です。テキスト領域に直接条件を入力できますが、**[条件の追加]** ボタンをクリックする方が簡単です。

**注:** 値と、表示フィールドに表示される値との違いは明確でないことがあります。多くの場合、それらは同じです。これらの違いは、表示フィールドはユーザーに表示される値であり、値は格納されている値です。例えば、通貨の値が小数点以下2桁で格納される場合、表示では、これらの数値は省略される可能性があります。この

違いは、ID フィールドでほぼ常に使用されます。それは、表示するのは名前にし、ユーザーが選択したときに、格納される実際の値としては ID を使用するのが有効なためです。また、コードがオブジェクト ID 以外のコードであるコードテーブルを使用することがよくあります。例えば、都道府県オブジェクトのコードが 2 文字の略語であり、他のオブジェクトの値として格納されるものの、ユーザーが都道府県を選択するときには、ユーザーに都道府県名を表示する場面があると考えてください。

このダイアログの使用後、ダイアログによって追加された情報は編集できます。

フィールドの内部名が何かが不明な場合があるため、選択リストからフィールドを選択することをお勧めします。ここで入力されるテキストは、標準 SQL WHERE 句です。また、このテキストには、WHERE キーワードは含まれません。

複数の条件を作成する場合、標準の AND 演算子と OR 演算子を使用できます。NOT 演算子を使用して、条件の意味を逆にすることもできます。

各条件に、次の 4 つの部分があります。

- オブジェクト内のフィールドの名前。この例では、このフィールドは、アカウントオブジェクト（[\[参照\]](#) 領域内で指定するオブジェクト）内に含まれているフィールドの 1 つです。
- 左側のフィールド値と右側の値を比較するときに Process Designer が使用する演算子。
- 比較されるデータの種類。選択肢は [コンテンツ]、[計算式]、または [フィールド] です。[\[フィールド\]](#) を選択した場合、現在のオブジェクトの最初のフィールド内のコンテンツと、クエリ対象のオブジェクトのタイプ内のフィールドのコンテンツが比較されます。
- フィールドが左側のデータが、比較対象のデータです。ソースが [コンテンツ] の場合、この領域内をクリックすると、右側に小さなアイコンが表示されます。これを選択すると、Process Designer に、比較の実行時に使用される別のオブジェクト内のフィールド名を選択できる選択リストが表示されます。

SQL LIMIT 句をクエリに追加するには、**Where** 句のテキストボックスに入力します。例えば、「Limit 200」と入力すると、取得した行数を 200 に制限できます。WHERE 情報の後にスペースを 1 つ追加し、SQL LIMIT を入力します。または、改行して SQL LIMIT 情報を入力します（「Limit」は大文字、小文字、または大文字と小文字を混在させて記述できます）。デフォルトでは、100 行が返されます。プロセスに 100 を超える行が必要な場合、この句を追加する必要があります。ただし、読み取り専用のオブジェクトリストには、関連付けられたクエリを 100 を超えた数に設定しても、100 行の制限があります。

JDBC 接続を使用する場合、ORDER BY 関数を使用してデータをソートできます。

ORDER BY 関数を使用するには、[\[ソート順\]](#) をクリックし、フィールドを選択します。例えば、名前のリストを人物の姓でソートできます。

**注:** ORDER BY 関数を使用を使用して、OData 対応の要求を JDBC コネクタに対して行うことはできません。

## リストを表示: フィールドの選択リスト

レコードタイプは、オブジェクトの一部を選択できるオブジェクトに関連付ける情報です。例えば、リードに「On the Road」レコードタイプと「In the Office」レコードタイプを設けるとします。プロセスは、これらの値を使用して、表示するその他の情報を選択できます。レコードタイプの値が「On the Road」の場合、A、B、C の値を持つ選択リストを設定します。値が「In the Office」の場合、C、D、E の値の選択リストを設定します。

依存関係選択リストを持つ作成サービスステップを使用する場合は、このオプションを使用します。[\[オブジェクトタイプ\]](#) と [\[フィールド名\]](#) が自動的に入力されるのに対し、[\[コントローラ値\]](#) は手動で入力する必要があります。ユーザーがレコードタイプを含むフィールドに入力する場合、[\[ソース\]](#) を [\[フィールド\]](#) に設定し、依存先の該当するフィールドも選択します。[\[コンテンツ\]](#) の場合、その値を入力します。

例えば、[\[事業部門\]](#) 選択リストと、それに依存する [\[業種\]](#) 選択リストを持つリードの作成サービスステップがあるとします。ユーザーが [\[事業部門\]](#) 選択リストから [\[テクノロジー\]](#) を選択すると、Process Designer はその部門に設定されている、[\[コンピュータハードウェア\]](#) や [\[バイオテクノロジー\]](#) などの値を [\[業種\]](#) 選

択リストに表示できます。これを [作成] 画面で正しく操作するには、[業種] フィールドのフィールドプロパティを編集し、[コントローラ値] を [フィールド] に変更し、フィールドとして [事業部門] を選択する必要があります。

[フィールド] に [選択リスト] を選択したときのオプションは次のとおりです。

- **オブジェクトタイプ:** オブジェクトフィールドを格納するオブジェクトタイプ。これはすでに正しく入力されているため、変更する必要はありません。
- **フィールド名:** 選択リストの値を定義するフィールドの名前。これはすでに正しく入力されているため、変更する必要はありません。
- **コントローラ値:** 依存する選択リストを正しく表示するには、このコントローラ値を入力します。この値をコンテンツに設定するのは適していません。これは、レコードタイプの ID は、サンドボックスや製品で使用されるときとは異なるためです。クエリを使用して、コントローラ値を動的に設定するのが適しています。このクエリは、1つの値のみを返すものである必要があります。

## リストを表示: ロール内のユーザー

**ロール内のユーザー**では、Process Designer に対して、識別するロールを持つすべてのユーザーを特定することを指定します。

[リストを表示] にこのオプションを選択するとき、そのロールのユーザーを見つけるために Process Designer が使用するロール名も選択します。これらの各のソースは、コンテンツ、フィールド、計算式、画面です。

## 式エディタの使用

式エディタを使用すると、プロセスまたはサービスコネクタを定義するときに、フィールド定義を編集できます。

式エディタは、Process Designer から次の2つのコンテキストで開けます。

- フィールド定義の [ソース] タイプとして [計算式] を選択すると、フィールド値を計算式から取得できません。
- サービスコネクタの URL、パラメータ、HTTP ヘッダーを定義するときに使用できます。

式エディタを使用すると、次の操作を行えます。

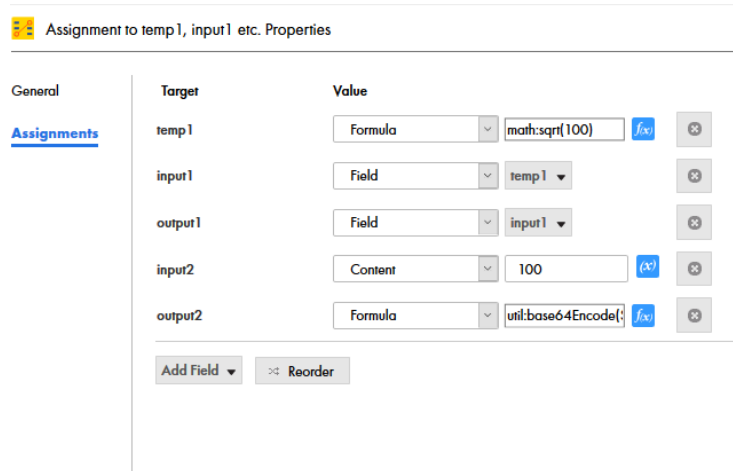
- XML および XQuery 用に強調表示された構文を参照できます。
- 構文の検証フィードバックを表示できます。
- 入力時に表示されるコード補完オプションを選択できます。
- よくあるコードのフラグメントをドロップダウンリストから挿入できます。
- キーボードショートカットを使用します。
- プロセス、現在のステップ、現在のプロセス内の任意のアップストリームのステップに対して定義済みの使用可能な入力、出力、一時の各フィールドのリストから、フィールドを選択します。フィールドを選択すると、該当する値（「\$input.Customer」など）がエディタに挿入されます。
- 使用可能な、カテゴリ別にグループ化された関数のリストから、関数を選択します。このリストには、いくつかのよくある XQuery 関数が含まれ、構文ヘルプがある場合はツールヒントとして表示されます。

- 選択した項目を表すサンプル XML（プロセスオブジェクト XML 要素またはプロセスオブジェクトの子の場合）を挿入します。
- よくあるコードのフラグメントをドロップダウンリストから挿入できます。

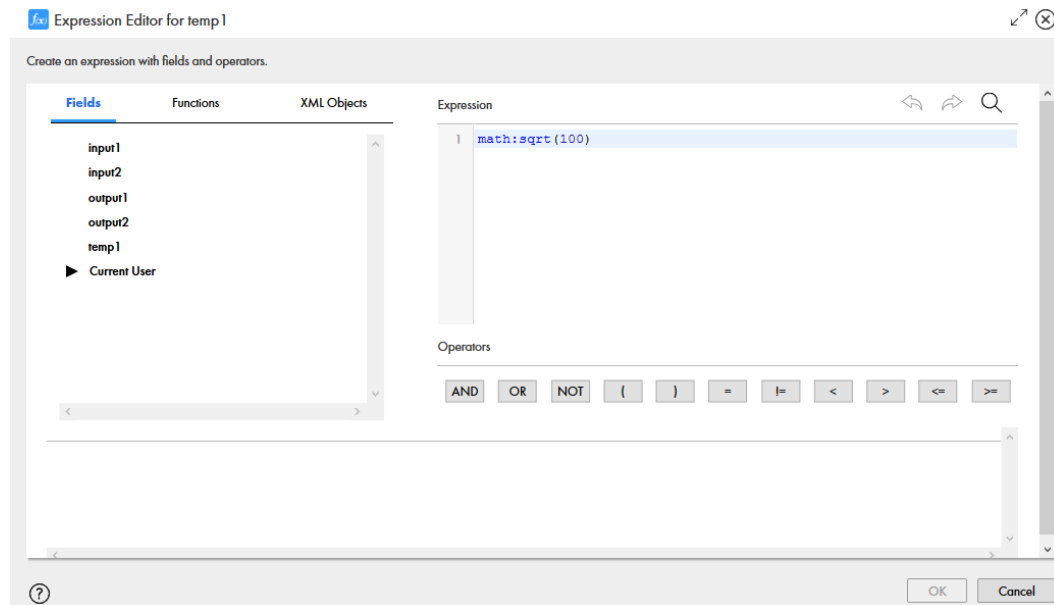
## 式エディタの起動

[f (x)] アイコンがある場合、これをクリックし、式エディタを開きます。

例えば、割り当てステップでは、計算式を持つ入力フィールドをデータソースとして追加することがあります。次の図に示すように、**[計算式]** をリストから選択できます。

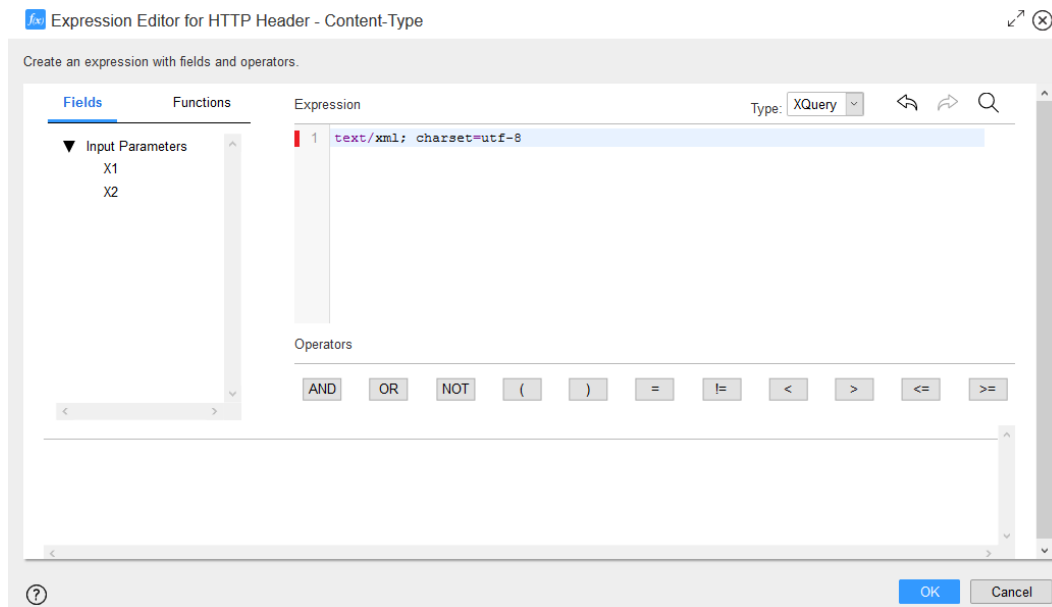


式エディタが開いたら、下記の機能を使用できます。次の図 (input.TestInput) に示すように、フィールドタイプと名前がウィンドウタイトルに表示されます。





サービスコネクタから開かれるときに、タイプ（XML、XQuery、JSON、コンテンツ）を選択し、図に示すように、接続プロパティと入力パラメータ内で使用できるフィールドを表示できます。



## オプションの編集

式エディタを使用するときに、ツールバーと、次のキーボードショートカットを使用できます。これらはエディタがアクティブなときに（カーソルの点滅中）使用できます。

元に戻す	Ctrl+Z
やり直し	Ctrl+Y
コピー	Ctrl+C
切り取り	Ctrl+X
貼り付け	Ctrl+V
検索	Ctrl+F
4スペース分をインデント	Tab
使用可能な変数の一覧を表示	\$
使用可能な挿入物（名前空間、関数、フィールド、よくあるコードフラグメント）のリストを表示	Ctrl+スペース (エディタ内で入力を開始し、リストをフィルタして、必要な関数や変数を選択することもできます)。

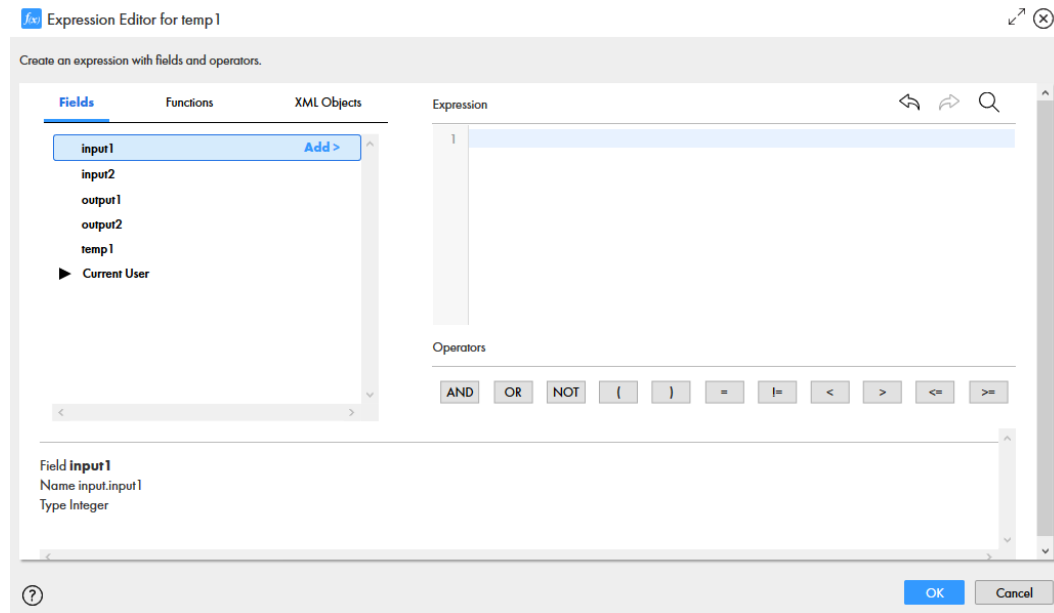
ツールバーアイコンを使用して、エディタウィンドウを全画面表示に拡張することもできます。Esc をクリックすると、全画面表示のエディタが閉じ、キャンバスに戻ります。

## 計算式の構築

選択したフィールドの値を特定する計算式を入力するには、ツールバーからこれらのオプションにアクセスできます。

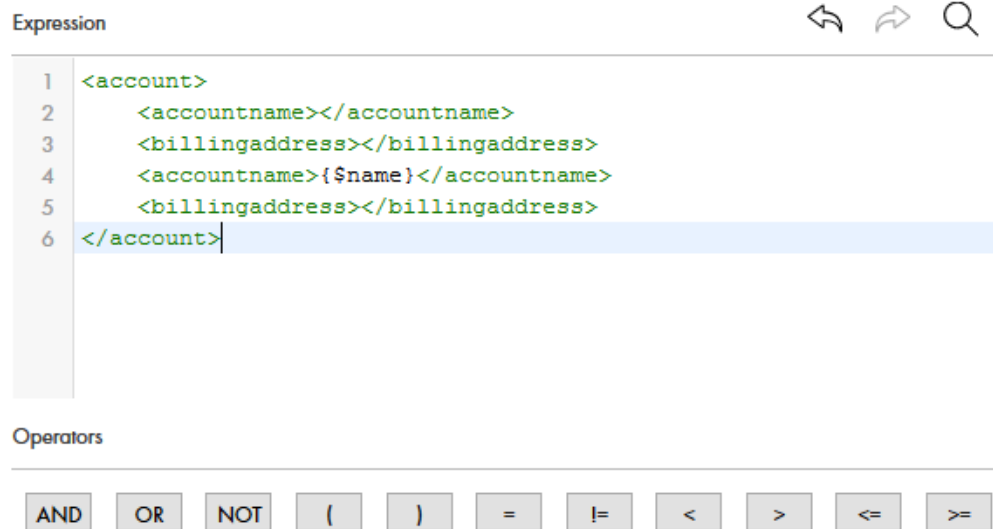
- フィールドの挿入。計算式に挿入できる使用可能なフィールドの一覧が表示されます。
- 関数の挿入。使用可能な関数の一覧が表示されます。
- XML オブジェクト。追加できる XML オブジェクトがある場合は、その一覧が表示されます。

例えば、次の図では、**[挿入]** 一覧に、オブジェクトと関連付けられたフィールド名が表示されています。

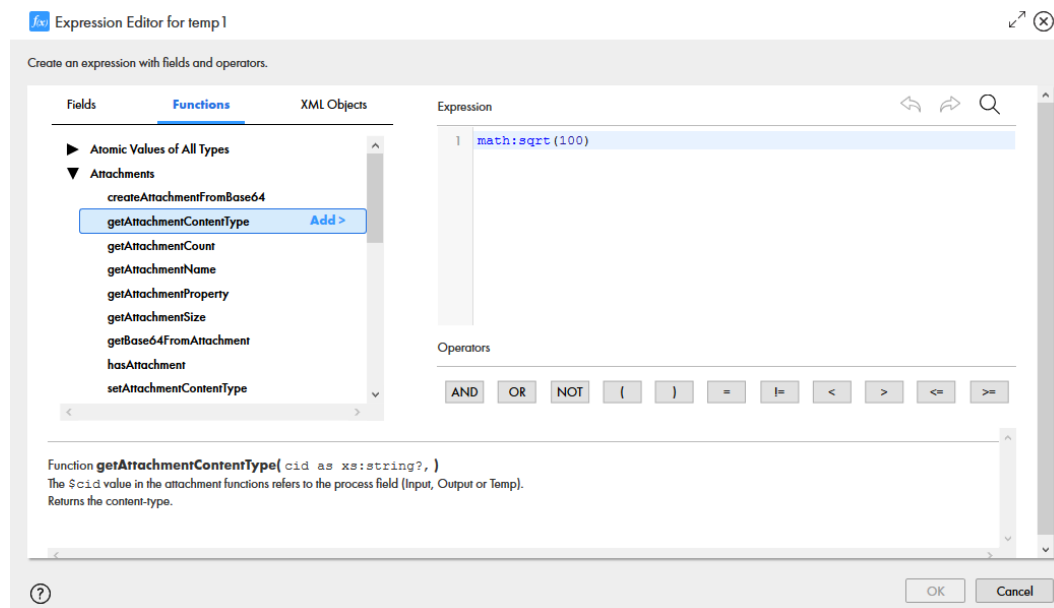


**[関数]** を選択し、カテゴリ別にソートされた使用可能な関数のリストを表示します。

**XML [オブジェクト]** リストを使用して、プロセスオブジェクトを定義するときに、フィールドに XML コードスニペットを追加します。例えば、プロセスオブジェクトを表す XML スニペットを作成し、図に示すように、[フィールドの挿入] を使用して、その入力フィールドを追加することができます。



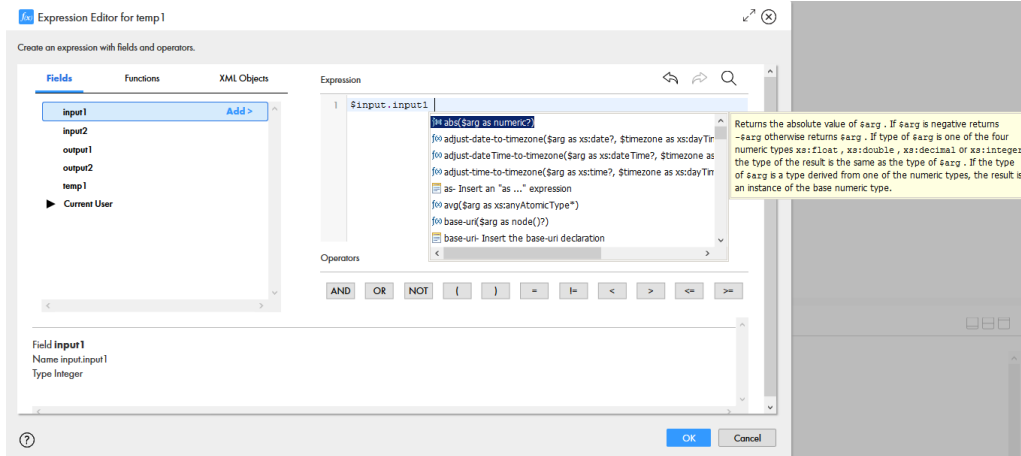
関数の詳細情報を参照するには、リスト内の関数名の上にカーソルを置き、説明を表示します。以下に例を示します。



エディタを使用するときは、次のヒントを考慮してください。

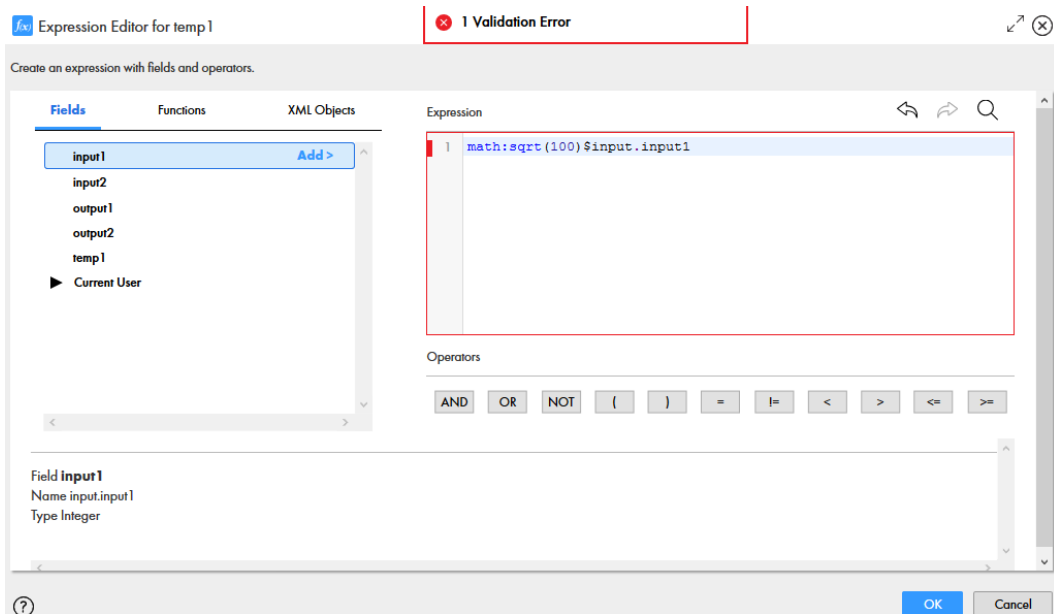
- **Ctrl+Space** を押すと、使用可能な挿入物（名前空間、関数、フィールド、よくあるコードフラグメント）のリストが表示されます。
- **\$** を押すと、使用可能な関数、プロセスフィールド、ローカルで宣言された任意の XQuery 変数のリストが表示されます。

- 式がXML内で使用される場合、前後に中括弧を追加することに注意してください。例:  
<Value>{2\*fn:abs(\$input.In)}</Value>
- リストをフィルタするには、いくつかの文字を入力します。



## 構文の検証

計算式を構築するときに、構文が検証されます。エラーが検出された場合、エラーがある行の横に赤いXが表示されます。そのエラーインジケータの上にカーソルを置くと、エラーの説明が表示されます。



構文を入力するにつれて構文が検証され、エラー状態が発生した場合、式エディタにメッセージが表示されます。

XQuery エラーメッセージは、XQuery エンジンによって定義されます。XQuery の検証の詳細と XQuery エラー状態（付録 F）のリストについては、次で参照できます: <http://www.w3.org/TR/xquery-3/>。

# 関数の使用

XQuery を Process Designer 内で操作する場合、または値を設定するために計算式を使用するフィールド内で操作する場合に、次の関数を使用できます。

計算式エディタ内で使用できる XQuery 関数の詳細については、ここではなく、<http://www.xqueryfunctions.com/xq/alpha.html> を参照してください。

式エディタを使用して関数を操作する方法については、「[式エディタの使用](#)」(ページ 23)を参照してください。

**注:** アプリケーション統合は、式エディタのインライン関数をサポートしていません。式エディタには、オペレーティングシステムの機能呼び出すことができる強力な関数が付属しています。関数を使用する前に、関数に渡されたコンテンツを確認する必要があります。

## すべてのタイプの原子値

プロセスおよびサービスコネクタでは、次の機能を使用できます。

- boolean
- empty
- exists
- false
- nilled
- not
- number
- string
- true

## プロセスおよびサービスコネクタの添付ファイル

次の表に示す関数は、プロセスおよびサービスコネクタ内で添付ファイルを処理するために使用できます。

プロセスの場合、添付ファイル関数内の \$cid 値は、プロセスフィールド (Input、Output、Temp) を参照します。例えば、input パラメータが、添付ファイルのサイズを取得するために customerPhoto (添付ファイルタイプ) を呼び出す場合、次の構文を使用します。

*sff:GetAttachmentSize(\$input.customerPhoto)*

**注:** フィールド名には、XQuery 内で簡単に参照できるように、スペースを含めないでください。

添付ファイル関数の使用方法のその他の例については、「[添付ファイル](#)」(ページ 44)を参照してください。

機能	構文	説明	可用性
base64EncodeAttachment	sff:base64EncodeAttachment(\$cid as xs:string?)	Base64 エンコードされた添付ファイルのコンテンツであることを示します。	プロセスおよびサービスコネクタ
createAttachmentFromBase64	sff:createAttachmentFromBase64 (\$contentName as xs:string, \$encodedContent as xs:string, \$mimeType as xs:string)	添付を Base64 エンコードされたコンテンツから作成します。	プロセスおよびサービスコネクタ

機能	構文	説明	可用性
getAttachmentCidByIndex	svc:getAttachmentCidByIndex(\$index as xs:integer?,)	応答添付ファイルの\$cid 値をインデックス別に返します。	サービスコネクタのみ
getAttachmentCidByName	svc:getAttachmentCidByName(\$name as xs:string?,)	応答添付ファイルの\$cid 値を名前別に返します。	サービスコネクタのみ
getAttachmentContentType	sff:getAttachmentContentType(\$cid as xs:string?)	Content-Type を返します。	プロセスおよびサービスコネクタ
getAttachmentCount	sff:getAttachmentCount(\$cid as xs:long)	添付ファイルの数を返します。	プロセスおよびサービスコネクタ
getAttachmentName	sff:getAttachmentName(\$cid as xs:string?)	添付ファイルがあれば、その名前を返します。	プロセスおよびサービスコネクタ
getAttachmentProperty	sff:getAttachmentProperty(\$cid as xs:string?, \$attribute as xs:string)	Content-Type などの MIME ヘッダー名を指定する添付ファイル属性を返します。	プロセスおよびサービスコネクタ
getAttachmentSize	sff:getAttachmentSize(\$cid as xs:long)	添付ファイルのサイズを返します。	プロセスおよびサービスコネクタ
getBase64FromAttachment	sff:getBase64FromAttachment(\$cid as xs:string)	Base64 エンコードされた添付コンテンツを変数（添付タイプ）から返します。	プロセスおよびサービスコネクタ
hasAttachment	sff:hasAttachment(\$cid as xs:boolean)	添付ファイルが存在するかどうかをチェックします。	プロセスおよびサービスコネクタ
setAttachmentContentType	sff:setAttachmentContentType(\$cid as xs:string, \$val as xs:string)	添付ファイルの Content-Type を設定します。	プロセスおよびサービスコネクタ
setAttachmentName	sff:setAttachmentName(\$cid as xs:string, \$val as xs:string)	添付ファイル名を設定します。	プロセスおよびサービスコネクタ
setAttachmentProperty	sff:setAttachmentProperty(\$cid as xs:string, \$attribute as xs:string, \$val as xs:string)	添付 MIME ヘッダー属性値を設定します。	プロセスおよびサービスコネクタ

## デジタル署名

次の項に示す関数は、デジタル署名内で使用できます。

## HMAC 関数

次の関数を使用することで、キー付きのハッシュメッセージ認証コード (HMAC) の計算によってデジタル署名を生成できます。

機能	構文	説明
hmacSignature	<code>dsig:hmacSignature( \$data as xs:string, \$key as xs:string, \$algorithm as xs:string, \$encoding as xs:string?) as xs:string</code>	指定したアルゴリズム (HMACSHA256、HMACSHA512 など) とオプションのエンコード (\$encoding は次のいずれかの値) を使用して、HMAC 署名を生成します。 <ul style="list-style-type: none"><li>- Base64 (デフォルト)</li><li>- Base64Url</li><li>- Hex</li><li>- Hex64</li></ul>
hmacSHA1signature	<code>dsig:hmacSHA1signature( \$data as xs:string, \$key as xs:string, \$encoding as xs:string?) as xs:string</code>	SHA1 アルゴリズムとオプションのエンコード (\$encoding は次のいずれかの値) を使用して、HMAC を計算します。 <ul style="list-style-type: none"><li>- Base64 (デフォルト)</li><li>- Base64Url</li><li>- Hex</li><li>- Hex64</li></ul> <p>\$data は Base64 エンコード内の JSON 文字列を含む 4 つの部分から構成されます。</p> <p>ペイロードでの hash:hash 関数の使用時に、Base64 でなく 16 進数のバイナリを確実に取得するには、次の式を使用します。</p> <pre>let \$md5hex := hash:hash(\$jsonPayload,"MD5") return xs:base64Binary(xs:hexBinary(\$md5hex))</pre>
hmacSHA256signature	<code>dsig:hmacSHA256signature( \$data as xs:string, \$key as xs:string, \$encoding as xs:string?) as xs:string</code>	SHA256 アルゴリズムとオプションのエンコード (\$encoding は次のいずれかの値) を使用して、HMAC を計算します。 <ul style="list-style-type: none"><li>- Base64 (デフォルト)</li><li>- Base64Url</li><li>- Hex</li><li>- Hex64</li></ul>
hmacSHA512signature	<code>dsig:hmacSHA512signature( \$data as xs:string, \$key as xs:string, \$encoding as xs:string?) as xs:string</code>	オプションのエンコード (\$encoding は次のいずれかの値) を使用して、HMAC SHA512 署名を生成します。 <ul style="list-style-type: none"><li>- Base64 (デフォルト)</li><li>- Base64Url</li><li>- Hex</li><li>- Hex64</li></ul>

## キー署名関数

次の関数を使用することで、プライベートキーに基づいて、デジタル署名を生成できます。

機能	構文	説明
signWithKeyFile	<code>dsig:signWithKeyFile( \$messageToSign as xs:string, \$pathToKey as xs:string, \$encryptionAlgorithm as xs:string, \$digestAlgorithm as xs:string, \$encoding as xs:string ) as xs:string</code>	<p>非対称アルゴリズムと、PKCS8 ファイル内で指定するプライベートキーを使用して、署名を生成します。</p> <p>引数:</p> <ul style="list-style-type: none"> <li>- \$digestAlgorithm: SHA1 または SHA256</li> <li>- \$encryptionAlgorithm: RSA (一般的) または DSA (\$digestAlgorithm に SHA1 を使用する場合)。</li> <li>- \$pathToKey: Base64 エンコードされた文字列としての PKCS8 証明書 (-----BEGIN PRIVATE KEY----- .....n-----END PRIVATE KEY-----\n) またはバイナリプライベートキーファイル。</li> <li>- \$encoding (省略可)。これは次のいずれかにできます。 <ul style="list-style-type: none"> <li>- Base64 (デフォルト)</li> <li>- Hex64</li> <li>- Base64Url</li> </ul> </li> </ul>
signWithKeyString	<code>dsig:signWithKeyString( \$messageToSign as xs:string, \$key as xs:string, \$encryptionAlgorithm as xs:string, \$digestAlgorithm as xs:string, \$encoding as xs:string ) as xs:string</code>	<p>非対称アルゴリズムと、PKCS8 証明書エンコードされた文字列内で指定するプライベートキーを使用して、署名を生成します。</p> <p>引数:</p> <ul style="list-style-type: none"> <li>- \$digestAlgorithm: SHA1 または SHA256</li> <li>- \$encryptionAlgorithm: RSA (一般的) または DSA (\$digestAlgorithm に SHA1 を使用する場合)。</li> <li>- \$key: Base64 エンコードされた文字列としての PKCS8 証明書 (-----BEGIN PRIVATE KEY----- .....n-----END PRIVATE KEY-----\n) またはバイナリプライベートキー。</li> <li>- \$encoding (省略可)。これは次のいずれかにできます。 <ul style="list-style-type: none"> <li>- Base64 (デフォルト)</li> <li>- Hex64</li> <li>- Base64Url</li> </ul> </li> </ul>
signWithCertificate	<code>dsig:signWithCertificate( \$messageToSign as xs:string, \$pathToCertificate as xs:string, \$keyRecoveryPassword as xs:string, \$encryptionAlgorithm as xs:string, \$digestAlgorithm as xs:string, \$encoding as xs:string, \$keyStorePassword as xs:string, \$aliasName as xs:string, \$keyStoreType as xs:string ) as xs:string</code>	<p>PKCS12 証明書を使用して、署名を生成します。</p> <p>引数:</p> <ul style="list-style-type: none"> <li>- \$pathToCertificate: PKCS12 または JKS (Java キーストア) のいずれかの証明書を格納するエージェント上のファイルの場所。</li> <li>- \$keyRecoveryPassword: 証明書内のキーにアクセスするためのパスワード。</li> <li>- \$keyStorePassword: キーストアを開くためのパスワード。空の場合、キーストアはパスワード保護されないと見なされます。</li> <li>- \$aliasName: 省略可。キーが格納された、Secure Agent のキーストア内のエントリのエイリアス。エイリアスが指定されない (または空) 場合、最初のエントリが使用されます。</li> <li>- \$keyStoreType: キーストアのタイプ。これは次のいずれかにできます。 <ul style="list-style-type: none"> <li>- PKCS12 (デフォルト)</li> </ul> </li> </ul>



機能	構文	説明
		- JKS (Java キーストア)

## ハッシュ関数

次の関数を使用することで、メッセージハッシュ文字列を生成できます。

機能	構文	説明
シャープ	hash:hash(\$string, \$alg)	指定したアルゴリズム（省略可）を使用して、メッセージのためのハッシュ文字列を生成します。ここで\$alg は次のいずれかにできます。 - MD5（デフォルト） - SHA1 - SHA256

## リスト関数

リスト関数は、ネイティブの Saxon 関数のみをサポートします。次の関数を使用することで、リストを操作できます。

機能	構文	説明
append	list:append(\$objectlist, \$newItem)	新規項目をリストに追加します。
count	list:count(\$objectlist)	リスト内の項目数を数えます。
head	list:head(\$objectlist)	リスト内の最初の項目を返します。
list	list:list(\$sequence)	ID のシーケンスをオブジェクトリスト用に ID のセミコロン区切りリストに変換します。
remove	list:remove(\$objectlist, \$position)	一覧から指定した位置の項目を削除します。
replace	list:replace(\$objectlist, \$position, \$newItem)	リスト内の既存項目を新しい値に置換します。
sequence	list:sequence(\$objectlistFieldName)	セミコロン区切りリストをシーケンスに変換します。オブジェクトリストフィールドが計算式およびコンテンツに挿入されるときに、フィールドは list:sequence(object_field_name) XQuery 関数内にラップされます。ホステッドオブジェクト内のオブジェクトリストの場合、これにより、オブジェクトリスト内の ID のセミコロン区切りリストはシーケンスに変換されます。プロセスオブジェクトのオブジェクトリストの場合、値はすでにシーケンスであり、関数によって返されるリストは変更されません。
tail	list:tail(\$objectlist)	リスト内の最初の項目以外の全項目を返します。

## リスト関数の例

最も一般的なリスト関数の使用例は、プロセスまたはサービスステップ内から、ステップが起動するごとに、毎回リストからオブジェクトを取得するものです。以下に例を示します。

```
LET mylist.Current:=list:head(mylist.List)
LET mylist.List:=list:tail(mylist.List)
```

次の例は、オブジェクトリストをセミコロン区切りリストに変換し、各項目についてこの処理を繰り返します。

```
for $objectid in list:sequence($objectlist)
...
```

次の例は、ホステッドオブジェクトに対し、ID のシーケンスをオブジェクトリスト用に ID のセミコロン区切りリストに変換します。プロセスオブジェクトのオブジェクトリストの場合、値のシーケンスを返します。

```
let $mergedObjectLists :=
  ( as:sequence($objectlist1), as:sequence($objectlist2) )
return list:list($mergedObjectLists)
```

## 数学

プロセス、サービスコネクタ内で、<http://www.w3.org/2005/xpath-functions/math> で説明されている数学関数を使用できます。

## 詳細設定

以下の各種関数を使用できます。

機能	構文	説明
base64Decode	util:base64Decode(data, charSet)	charSet 引数で指定された文字セットに基づいて、指定された入力文字列の Base64 デコードされたバージョンを返します。この関数は通常、添付ファイルに使用されます。 アプリケーション統合は、Azul JDK でエンコード用にサポートされる文字セットをサポートします。これには、US-ASCII、ISO-8859-1、UTF-8、UTF-16BE、UTF-16LE、および UTF-16 などが含まれます。 charSet 引数のデフォルトは UTF-8 です。
base64Encode	util:base64Encode(data, charSet)	charSet 引数で指定された文字セットに基づいて、指定された入力文字列の Base64 エンコードされたバージョンを返します。 アプリケーション統合は、Azul JDK でエンコード用にサポートされる文字セットをサポートします。これには、US-ASCII、ISO-8859-1、UTF-8、UTF-16BE、UTF-16LE、および UTF-16 などが含まれます。 charSet 引数のデフォルトは UTF-8 です。
base64EncodeURL	util:base64EncodeUrl(str, charSet)	指定文字列の Base64 エンコードされたバージョンを返します。URL 内で安全に使用できます。 「+」文字と「/」文字がある場合、「-」と「_」に置き換えられます。「=」文字がある場合は削除されます。 charSet 引数のデフォルトは UTF-8 です。
error	fn:error	カスタムエラーを発行します。

機能	構文	説明
exactly-one	fn:exactly-one	シーケンスに正確に 1 つの項目のみが含まれている場合、そのシーケンスを返します。それ以外の場合、エラーを返します。
format	util:format(value, pattern, timezoneId, locale)	<p>文字列コンテンツをフォーマットします。次の引数を使用します。</p> <ul style="list-style-type: none"> <li>- value: フォーマットされる文字列。</li> <li>- pattern: 値のフォーマット方法を示すパターン。<a href="#">「日付、時刻、番号の形式設定」(ページ 19)</a>を参照してください。</li> <li>- timezoneId: 省略可。タイムゾーン ID。格納値を使用しない場合 (通常は、先に <code>date:getTimeZone()</code> を呼び出すことによって返されます)、Process Designer が <code>date:getTimeZone()</code> を呼び出すので、この引数は省略できます。</li> <li>- locale: 省略可。ロケール。格納値を使用しない場合 (通常は、先に <code>date:getLocale()</code> を呼び出すことによって返されます)、Process Designer が <code>date:getLocale()</code> を呼び出すので、この引数は省略できます。</li> </ul> <p>locale 引数を使用し、timezoneId 引数を使用しない場合、timezoneId の後に来るコンマを追加する必要があります。以下に例を示します。</p> <pre>util:format("789", "##00.00", "", date:getLocale())</pre> <p>パターン引数に日付が含まれている場合は、小文字を使用して年形式を示します。例えば、yyyy を使用します。大文字を使用すると、エラーが発生する可能性があります。</p>
generate-random-string	util:generate-random-string(length as xs:integer)	指定した長さのランダム文字列を生成します。
generateUUID	util:generateUUID()	ユニバーサルに一意的な識別子を生成します。
getAssetLocation	util:getAssetLocation()	関数を使用するプロセスまたはガイドが保存されている場所を返します。詳細については、 <a href="#">「getAssetLocation」(ページ 40)</a> を参照してください。
getAssetName	util:getAssetName()	関数を使用するプロセスまたはガイドの名前を返します。詳細については、 <a href="#">「getAssetName」(ページ 41)</a> を参照してください。
getCatalogResource	util:getCatalogResource()	<p>組織内のカタログの場所 URL に基づくリソースを返します。Informatica Cloud 組織内のカタログリソースをパラメータとして取得し、要素ノードを返します。</p> <p>要素データを表示するには、ラッパー関数を追加します。以下に例を示します。</p> <pre>serialize(util:getCatalogResource("project:/spi.ipd/services.xml"))</pre>

機能	構文	説明
getInstanceStartTime	util:getInstanceStartTime()	指定されているプロセスまたはガイドの実行中インスタンスの開始時刻を返します。詳細については、 <a href="#">「getInstanceStartTime」 (ページ 41)</a> を参照してください。
getOrganizationId	util:getOrganizationId()	現在実行中のプロセスのコンテキストの組織 ID を示す文字列を返します。詳細については、 <a href="#">「getOrganizationId」 (ページ 40)</a> を参照してください。
getOrganizationName	util:getOrganizationName()	実行中のプロセス、ガイド、サービスコネクタ、またはデータアクセスサービスコネクタのコンテキストで組織名を返します。詳細については、 <a href="#">「getOrganizationName」 (ページ 40)</a> を参照してください。
getProcessId	util:getProcessId()	現在実行中または完了済みのプロセスのプロセス ID を返します。
getUserName	util:getUserName()	プロセスを実行している認証ユーザーのログイン名または ID である文字列を返します。
getUserSystem	util:getUserSystem()	プロセスを実行しているユーザーを認証したシステムの名前を表す文字列を返します。
one-or-more	fn:one-or-more	シーケンスに 1 つ以上の項目が含まれている場合、そのシーケンスを返します。それ以外の場合、エラーを返します。
parseJSON	util:parseJSON(jsonStr)	指定の JSON 文字列を解析し、これを XML 要素に変換します。
parseXML	util:parseXML(xmlStr)	指定の XML 文字列を解析し、これを XML 要素に変換します。
random	util:random()	0 から 1 の範囲の乱数を返します。
resolveURN	util:resolveURN()	組織の URN マッピングを取得します。
safeNumber	util:safeNumber()	数値を安全に格納して、変更できないようにします。
setProcessTitle	ipd:setProcessTitle()	プロセスのタイトルを設定します。 <b>注:</b> この関数は、Informatica Process Designer 内でのみ使用します。
toJSON	util:toJSON(elements)	指定の XML 要素のリストを JSON 文字列に変換します。
toXML	util:toXML(element)	指定の XML 要素を XML 文字列に変換します。

機能	構文	説明
trunc	util:trunc(arg, precision)	精度で指定された小数点以下の桁数に基づいて、数値を特定の桁に切り詰めます。詳細については、「 <a href="#">trunc (Numbers)</a> 」 ( <a href="#">ページ 39</a> )を参照してください。
zero-or-one	fn:zero-or-one(arg)	シーケンスに 0 または 1 つの項目が含まれている場合、そのシーケンスを返します。それ以外の場合、エラーを返します。

## 数字

式エディタでは、次の数字関数を使用できます。

- abs
- avg
- ceiling
- floor
- max
- min
- round
- round-half-to-even
- sum

## シーケンス

式エディタでは、次のシーケンス関数を使用できます。

- count
- distinct-values
- index-of
- insert-before
- last
- position
- remove
- reverse
- subsequence
- unordered

## 文字列

式エディタでは、次の文字列関数を使用できます。

- codepoint-equal
- codepoints-to-string
- compare
- concat

- contains
- default-collation
- ends-with
- lang
- lower-case
- matches
- normalize-space
- normalize-unicode
- replace
- starts-with
- string-join
- string-to-codepoints
- substring
- substring-after
- substring-before
- tokenize
- translate
- upper-case

## XML

式を構築するときに、共通 XML 関数を追加できます。式エディタでは、共通 XML 関数のリストが以下のカテゴリに表示されます。

- XML ドキュメント、URI、ID（プロセスのみ）
- XML 名前空間と名前
- XML ノード

## その他の関数

次の関数は、式エディタの [その他] セクションで使用できます。

## trunc (Numbers)

精度で指定された小数点以下の桁数に基づいて、数値を特定の桁に切り詰めます。

### 構文

```
util:trunc(arg, precision)
```

引数	必須/ オプション	説明
<i>arg</i>	必須	数値データタイプ。切り詰めを行う引数を渡します。数値データ型と評価される有効な式を入力します。
<i>precision</i>	オプション	正または負の整数です。整数を求める有効な式を必要に応じて入力できます。この整数は、切り詰め後の桁数を指定します。

*precision* が正の整数である場合、trunc は *precision* で指定された小数点以下の桁数を含む *arg* を返します。*precision* が負の整数である場合、trunc は小数点の左側の指定された桁数をゼロに変更します。*precision* 引数を省略すると、trunc は *arg* の小数部分を切り詰めて、整数を返します。

引数の小数点以下の末尾のゼロはすべて切り詰められます。例えば、次の式は結果として 2345.7535 を返しません。

```
util:trunc(2345.75350000, 6)
```

小数点以下が 16 桁を超える数値が含まれている場合、XQuery の制限により、結果は指数値を返します。例えば、次の式は結果として 1.234567812345679E7 を返します。

```
util:trunc(12345678.12345678901234567890, 15)
```

10 進数の *precision* 値を渡すと、*arg* を近似値の整数に四捨五入してから、式が評価されます。

### 戻り値

関数で指定されたパラメータに基づく数値または整数値。

### 例

以下の式は、PRICE カラム内の値の切り詰めを行います。

```
util:trunc(PRICE, 3)
```

PRICE	RETURN VALUE
12.9995	12.999
-18.8652	-18.865
56.9563	56.956
15.9928	15.992

```
util:trunc(PRICE, -1)
```

PRICE	RETURN VALUE
12.99	10
-187.86	-180
56.95	50

PRICE	RETURN VALUE
-------	--------------

1235.99	1230
---------	------

util:trunc(PRICE )

PRICE	RETURN VALUE
-------	--------------

12.99	12
-------	----

-18.99	-18
--------	-----

56.95	56
-------	----

15.99	15
-------	----

## getOrganizationId

実行中のインスタンスのコンテキスト内にある *Informatica Intelligent Cloud Services* 組織 ID の文字列を返します。

### 構文

```
util:getOrganizationId()
```

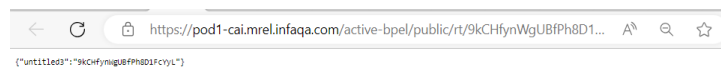
### 戻り値

プロセスがエージェントにデプロイされている場合、関数は出力を \$public として返します。

プロセスがクラウドサーバーにデプロイされている場合、関数は *Informatica Intelligent Cloud Services* 組織 ID の文字列を出力として返します。

### 例

次のサンプルは、関数の結果を示しています。



## getOrganizationName

実行中のプロセス、ガイド、サービスコネクタ、またはデータアクセスサービスコネクタのコンテキストで組織名を返します。

### 構文

```
util:getOrganizationName()
```

### 戻り値

この関数は、出力として組織名の文字列を返します。

## getAssetLocation

関数を使用するプロセスまたはガイドが保存されている場所を返します。

**注:** ガイドに加えて、getAssetLocation 関数はクラウドサーバー上で実行されるプロセスでのみ使用できます。

### 構文

```
util:getAssetLocation()
```



## 戻り値

この関数は、アセットが存在するフォルダの場所を返します。

## getAssetName

関数を使用するプロセスまたはガイドの名前を返します。

### 構文

```
util:getAssetName()
```

## 戻り値

この関数は、出力でアセット名を文字列として返します。

## getInstanceStartTime

指定されているプロセスまたはガイドの実行中インスタンスの開始時刻を返します。

### 構文

```
util:getInstanceStartTime()
```

## 戻り値

この関数は、指定されたアセットに対応する実行中のインスタンスの開始時刻を返します。この関数は出力を yyyy-MM-dd'T'HH:mm:ss.SS'Z' 形式で返します。

### 例

次のサンプルは、プロセスで使用される関数の結果を表示します。

```
2023-09-20T09:02:50.774Z
```

## 関数と変数の指定

サービスコネクタを構成するには、バインディング、出力フィールド、その他のプロパティを定義するための変数と関数を指定する必要があります。

### ビルトイン変数

これらの変数は、XQuery 式を使用してサービスコネクタで参照できます。

変数	変数型	説明
<code>\$VariableName</code>	すべての接続プロパティ	プロパティ、入力、その他のパラメータは、この形式を使用して指定できます。
<code>\$ResponseStatusCode</code>	出力フィールドマッピング	HTTP 応答コード。

変数	変数型	説明
\$ResponseHeaders	出力フィールドマッピング	各項目が次のとおりの要素リスト内に HTTP 応答ヘッダーが格納されます。 <pre>&lt;header name="Content-Type"&gt;text/plain&lt;/header&gt;</pre> 以下に例を示します。 <code>\$ResponseHeaders[@name="Content-Type"]/text()</code>
\$RESTResponse	出力フィールドマッピング	ヘッダーとコードを含む RESTResponse XML データが格納されます。サービスコネクタのテスト結果内に表示されます。

## 出力フィールドマッピング関数

サービスコネクタでバインディングを定義するときは、式エディタで多くの関数を使用できます。

使用可能なすべての機能については、[「関数の使用」 \(ページ 29\)](#)および[「式エディタの使用」 \(ページ 23\)](#)を参照してください。

次の表に、サービスコネクタの出力フィールドのマッピングで使用できる関数を示します。

機能	構文	説明
responseHeaderExists	<code>svc:responseHeaderExists(\$ResponseHeaders, headerName) : boolean</code>	ヘッダーパラメータが応答内にあるかどうかを示す Boolean 値を返します。
getResponseHeader	<code>svc:getResponseHeader(\$ResponseHeaders, headerName, defaultValue) : string</code> <code>svc:getResponseHeader(\$ResponseHeaders, headerName) : string</code>	応答ヘッダー値を返します。ヘッダーパラメータが定義されていない場合、この関数は、オプションのデフォルト値を返します。
getResponseHeaderNames	<code>svc:getResponseHeaderNames(\$ResponseHeaders) : list of strings</code>	応答ヘッダー内のすべてのパラメータの名前を格納したリストを返します。

## デジタル署名関数の概要

デジタル署名関数は、Process Designer に対して使用できます。XQuery を使用して、コンテンツの署名を処理するプロセスおよびサービスコネクタを作成できます。サポート対象の関数は次のいずれかです。

- 対称: ハッシュベースのメッセージ認証コード (HMAC) を使用して署名されます。このコードは、共有パスワードとキーに基づき、複数のキー形式を使用できます。
- 非対称: プライベートキー (PKCS8、PKCS12、Java KeyStore など) を使用して署名されます。

署名付きデジタルコンテンツのよくある用途の 1 つは、サービスコネクタ内での API 認証を可能にすることです。これにより、文字列ベースのコンテンツに対してハッシュと関連関数を使用できます。例えば、署名付きヘッダーをペイロードの一部として使用する必要がある場合があります。

各関数の詳細については、「[関数の使用](#)」(ページ 29)を参照して下さい。

**注:** 現在、署名付きコンテンツの検証またはバイナリコンテンツに署名するためにサポートされている関数はありません。

## HMAC

HMAC 署名方法は、次のサービスを含む多くのサービスで使用されます。

- Amazon Web サービス (AWS)
- Twitter OAuth

Process Designer は、HMAC SHA1、HMAC SHA256、HMAC SHA512 を Base64、Hex、Hex64、Base64Url のいずれかのエンコードでサポートしています。例えば、AWS 用のコンテンツに署名するために、これらのいずれかの関数を使用出来ます。

- `dsig:hmacSignature( $data as xs:string, $key as xs:string, $algorithm as xs:string, $encoding as xs:string? ) as xs:string`
- `dsig:hmacSHA1signature( $data as xs:string, $key as xs:string, $encoding as xs:string? ) as xs:string`
- `dsig:hmacSHA256signature( $data as xs:string, $key as xs:string, $encoding as xs:string? ) as xs:string`
- `dsig:hmacSHA512signature ( $data as xs:string, $key as xs:string, $encoding as xs:string?) as xs:string`

## AWS 認証

AWS REST API には、次の形式で認証ヘッダーが必要です。

Authorization: AWS AWSAccessKeyId:Signature

ここで、

`Signature = dsig:hmacSHA1signature($strToSign, "AwsSecretKeyId")`

AWSAccessKey および AwsSecretKeyId は、AWS によって生成され、AWS 開発者コンソールを介してアクセス出来ます。

**注:** \$strToSign は HTTP ヘッダー値とその他のパラメータの組み合わせです。

AWS のための REST 認証の詳細については、

<http://docs.aws.amazon.com/AmazonS3/latest/dev/RESTAuthentication.html> を参照して下さい。

## 非対称プライベートキーベースの署名

次のプライベートキーの方法がサポートされています。

- PKCS8
- PKCS12
- Java KeyStore

**注:** プライベートキーを使用してコンテンツに署名する場合、キーストアファイルは Secure Agent にあり、プロセスまたはサービスコネクタは、そのキーにアクセスするために、Secure Agent 上で実行されている必要があります。そのキーへのファイルパスが、オブジェクトプロパティに指定されている必要があります (C:/certs/mykey.p12 など)。

次の構成にすることも出来ます。

- PKCS8/12 アーティファクトをエージェントコントリビューションにデプロイし、「project:/」スキーマを使用して場所を指定します。
- ファイルの場所でなく、エンコードされたキーコンテンツを使用して、キーを提供します。この方法は、サービスコネクタの開発中に有益な場合があります。例えば、テスト目的で、エンコードされたコンテンツをテキスト領域に貼り付けられます。

## PKCS8 キーファイル

PKCS8 キーファイルを使用して署名するには、この関数を使用します。

```
dsig:signWithKeyFile( $messageToSign as xs:string, $pathToKey as xs:string, $encryptionAlgorithm as xs:string, $digestAlgorithm as xs:string ) as xs:string
```

キーファイルはバイナリまたは Base64 エンコード形式に出来ます。

## P12 およびキーストアベースのファイル

P12 またはキーストアベースのファイルを使用して署名するには、この関数を使用します。

```
dsig:signWithCertificate( $messageToSign as xs:string, $pathToCertificate as xs:string, $keyRecoveryPassword as xs:string, $encryptionAlgorithm as xs:string, $digestAlgorithm as xs:string ) as xs:string
```

例えば、サービスアカウントに対してこの Google JWT (JSON Web トークン) 認証用の方法を使用出来ます。Google プライベートキーを Secure Agent でアクセスして保存するために取得したら、JWT 要求を作成し、自分のプライベートキーを使用してそれに署名します。

JWT 認証の詳細については、<https://console.developers.google.com> および <https://developers.google.com/identity/protocols/OAuth2ServiceAccount> を参照して下さい。

## 添付ファイル

Process Designer はエンドポイントを公開することで、イベントハンドラ、内部 API 呼び出し、カスタムサービスコネクタ、Amazon S3 コネクタなどの組み込みコネクタを使用して、プロセス内で添付ファイルにアクセスできるようにします。これにより、応答内で関数を使用して (サポートされている場合) 添付ファイルを渡し、添付ファイルに関するファイルサイズ、ファイル名などのデータを抽出できます。

プロセスの設計時に、添付ファイルをサブプロセス、サービスコネクタ、サービスの各ステップに渡すか、添付ファイルを返せます (出力フィールドが添付ファイルのタイプを使用して定義されている場合)。

プロセス内では、添付ファイルが画面からの直接入力として入力される 1 つまたは複数の添付ファイルを追加することを、ユーザーに許可できます。これは、プロセスフィールドを使用してステップ間を移動する方法ではありません。例えば、プロセスを使用して、不動産データをキャプチャし、ユーザーが関連不動産の写真を添付するようになります。ただし、添付ファイルデータには、プロセス内でアクセスすることはできません。

**注:** プロセスを Cloud Server 上で実行する場合、サイズがデフォルトの 5,242,880 バイトを超える添付ファイルは使用しないでください。クラウドサーバーは、5,242,880 バイトを超える添付ファイルを処理できません。

## エンコード

添付ファイルは次の形式でエンコードできます。

- multipart/form-data (ブラウザベースの処理用)
- multipart/mixed

## データタイプ

2 つの簡単なデータタイプを使用して、名前付き入力パラメータを添付ファイルとして指定できます。

- **添付**は単一の添付ファイルをサポートし、フィールドプロパティとして最大ファイルサイズを定義できます。
- **添付ファイル**は複数の添付ファイルをサポートし、フィールドプロパティとして最大ファイルサイズと最大ファイル数を定義できます。

プロセス内から、XQuery 式と組み込み関数を使用して、添付ファイルに関するメタデータにアクセスできます。例えば、次の式では、写真入力のサイズが返されます。

```
sff:getAttachmentSize($input.photo)
```

添付フィールド割り当てステップを使用することで、出力フィールドまたは一時フィールドに入力添付フィールドを割り当てられます。その場合、ターゲットの一時フィールドまたは出力フィールドも [添付] タイプである必要があります。

この例では、2 番目のパラメータ (\$encodedContent) は、Base64 でエンコードされた文字列です。

```
sff:createAttachmentFromBase64("base64", "U0pGUIRA", "ascii")
```

**注:** プロセス内で添付関数を使用できますが、サービスコネクタ内では使用できません。サービスコネクタ内の添付ファイルの処理に関する詳細は、次を参照してください。

添付ファイルと一緒に使用できる関数の詳細については、「[関数の使用](#)」 ([ページ 29](#)) を参照してください。

## 添付ペイロード

単純な添付ファイルを使用する場合、次の状態であることを確認してください。

- 各名前付き部分は、プロセス内の入力パラメータと同じ名前 (名前とコンテンツの破棄) である必要があります。
- Content-Type 部分も、入力パラメータのコンテンツタイプと一致している必要があります。添付ファイル以外のすべてのデータタイプについて、Content-Type= text/plain (テキスト、チェックボックス、日付など) である必要があります。
- その部分がプロセスオブジェクトを表す場合、そのコンテンツは有効なシリアル化された JSON コンテンツ (Content-Type が application/json のもの) である必要があります。
- 添付タイプまたは添付ファイルタイプのプロセス出力フィールドを定義する必要があります。これらのフィールドは、呼び出し元に添付ファイルとして返されます。

例えば、プロセスに「first」(テキスト)、「last」(テキスト)、「inputFile」(添付) という入力フィールドがある場合、ペイロードは次のようになります。

```
POST /active-bpel/public/rt/Attachments_Test HTTP/1.1
Content-Type: multipart/form-data; boundary=----TheBoundary1234
-----TheBoundary1234
Content-Disposition: form-data; name="first"
John
-----TheBoundary1234
Content-Disposition: form-data; name="last"
Smith
-----TheBoundary1234
Content-Disposition: form-data; name="inputFile"; filename="filename.png"
Content-Type: image/png
... binary image data ...
-----TheBoundary1234--
```

応答の最初の部分は、応答出力フィールドを記述した JSON ドキュメントである必要があります。JSON コンテンツには、出力フィールド値 (標準応答) があります。

出力フィールドのタイプが添付の場合、その値は、添付を含む部分のコンテンツ ID (cid) を含む文字列になります。例えば、入力フィールドに「first」(テキスト)、「last」(テキスト)、「inputFile」(添付) という出力フィールドがある場合、応答は次のようになります。

```
Content-Type: multipart/mixed; boundary=----TheBoundary1234
-----TheBoundary1234
Content-Type: application/json
{
  "first" : "John",
  "last" : "Smith",
  "inputFile" : "cid:fc1c6030-4b90-4981-b9d8-ab1d0ba0e84e"
}
-----TheBoundary1234
Content-Type: image/png
Content-Name: photo
Attachment-Created-At:1435182142837
Content-Id: fc1c6030-4b90-4981-b9d8-ab1d0ba0e84e
```

```
Content-Length: 1073
... binary image data ...
```

```
-----TheBoundary1234--
```

- プロセスプロパティ内で**出力フィールドにはペイロード全体**を有効にせず、添付タイプの単一の出力フィールドがある場合、ここに示すように、応答には添付データとメタデータの両方が含まれます。

```
--Boundary
content-type: ...
```

```
cid:outputFile
```

```
--Boundary
content-type: image/jpeg
```

```
[binary data]
```

**出力フィールドにはペイロード全体**を有効にし、添付タイプの1つの出力フィールドのみがある場合、メタデータなしで、単一の添付（テキスト型またはバイナリ型）データを受け取れます。サードパーティが添付にのみアクセスする場合、コンシュームするほうが簡単です。以下に例を示します。

```
HTTP/1.1 200 OK
Content-Type: image/jpeg
Content-Length: length
```

```
[.. JPEG data ...]
```

## サービスコネクタ内での添付ファイルの使用

サービスコネクタは、プロセスからサービスコネクタの入力に添付を送信する、サービスコネクタから添付を取得してプロセスに添付として送信するなどの目的で定義できます。

添付ファイルはサービス入力（Salesforce ベースのサービスを含む）に対してプロセスとサービスコネクタから直接送信されます。

**注:** Amazon S3 などのサービスからは、コンテンツをアップロードまたはダウンロードし、出力マッピングで定義された名前でも添付をメインプロセスにプロパゲートして戻します。Amazon S3 コネクタを使用すると、S3 オブジェクトのコンテンツを Base64 でエンコードされた文字列として、または添付ファイルとして読み書きできます。詳細については、Amazon S3 コネクタに関するガイドを参照してください。

## サービスコネクタでの複数の出力添付ファイルの使用

サービスコネクタでは、マルチパート応答を適切に処理し、さまざまな応答パートに対し複数の添付ファイルを作成できます。パート関連メタデータ情報は、添付ファイルのプロパティで使用できます。

サービスコネクタでは、ペイロードパートも認識されます。最初の JSON/XML/テキストパートがペイロードとして使用されます。非マルチパート応答の使用時に通常行うアクションの出力フィールド式のペイロードからデータを抽出できます。選択されたペイロードは添付として使用できません。

**[出力]** タブで、**[添付ファイルの取得]** オプションを使用して複数の添付ファイルを処理し、添付ファイルの全リストをペイロードを除く選択された変数に渡すことができます。

サービスコネクタで添付ファイルを処理する式を使用することもできます。式エディタを開くと、**[出力添付ファイル]** セクションの **[関数]** タブで新しい関数を見つけることができます。

## GET、POST、PUT の使用

JSON または XML をサポートしていないリソースに対して GET、POST、PUT の各操作を使用するサービスコネクタを定義できます。

以下の点に注意してください。

- 入力パラメータ（添付タイプ）は、バインディング URL（POST/PUT）から宛先にアップロードする必要があります。バインディングタイプと添付ファイル数に基づくペイロード形式を使用します。

- 複数の添付パラメータがあり、バインディングが FORM の場合、すべてのパラメータ（非ファイルパラメータを含む）を multipart/form-data として送信します。
- 1つ以上の添付パラメータがあり、バインディングが JSON の場合、ファイルを multipart/form-data として送信し、非ファイルパラメータを単一の JSON 要素に組み込みます。
- 厳密に1つのファイルパラメータがあり、バインディングが CUSTOM であり、バインディングペイロードは空である場合（カスタムのテキスト領域内に XML または JSON なし）、Raw ファイルデータをそのまま POST または PUT します。
- 複数の添付パラメータがあり、バインディングが CUSTOM であり、ペイロードが XQuery の場合、HTTP 送信ペイロードは multipart/related である必要があります。最初の部分は CUSTOM (XQuery) コンテンツであり、各ファイル/添付入力パラメータの部分がこれに続きます。
- 添付フィールド割り当てステップを使用することで、出力フィールドまたは一時フィールドに入力添付フィールドを割り当てられます。その場合、ターゲットの一時フィールドまたは出力フィールドも [添付] タイプである必要があります。
- 複数のファイルのダウンロード（multipart/mime 添付の使用）はサポートされていません。

注: 「sff」関数はプロセス内でのみ使用できます。

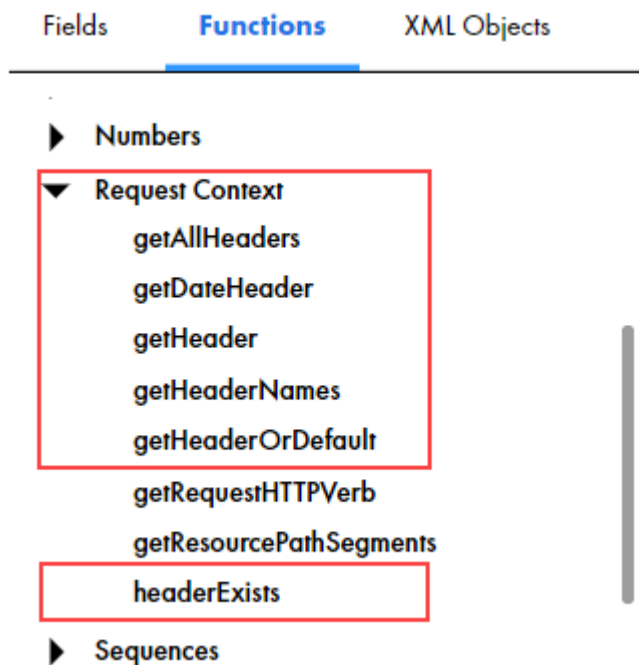
## HTTP および JMS ヘッダー

ヘッダー関数を使用して、プロセスの起動時に渡すヘッダーに関する詳細を取得します。プロセスヘッダーとメッセージイベントヘッダーの詳細を取得できます。

ヘッダー関数を使用して、プロセスの起動時に渡すヘッダーのみから入手できる情報を取得します。例として、日付、JSON Web トークン、セキュリティ ID などがあります。

フィールドにヘッダー値を割り当て、必要に応じて値に XQuery 操作を実行できます。

次の図は、式エディタの【要求コンテキスト】セクションで利用できるヘッダー関数を示します。



各関数の働きを理解するために、次のヘッダーと値で起動するプロセスについて検討します。

ヘッダー	値
accept-language	en-US;q=0.9
client-ip	192.0.2.1
accept	text/xml
postman-token	0023-a5dfd-8vdg3-n2b3
CustomHeader	This is a custom header
ExecutionDate	Wed, July 25 2018 06:25:24 GMT

次のヘッダー関数を単独で、または他の関数と一緒に使用できます。

### 全ヘッダー値の取得

**getAllHeaders** 関数を使用して、全ヘッダーとそれらの値のリストを取得します。

例えば、次の式を使用すると、特定の出力が得られます。

式:

**fn:getAllHeaders()**

出力:

```
<headers>
<header name="accept-language">en-US;q=0.9</header>
<header name="client-ip">192.0.2.1</header>
<header name="accept">text/xml</header>
<header name="postman-token">0023-a5dfd-8vdg3-n2b3</header>
<header name="CustomHeader">This is a custom header</header>
<header name="ExecutionDate">Wed, July 25 2018 06:25:24 GMT</header>
</headers>
```

### 日付ヘッダーの取得

**getDateHeader** 関数を使用して、日付ヘッダーの値を取得します。関数は、dateTime 型のデータを返します。

他の日付と時間の関数を使用して、値を解析できます。

例えば、次の式を使用すると、特定の出力が得られます。

式:

**fn:year-from-dateTime(request:getDateHeader("ExecutionDate"))**

出力:

2018

### 特定のヘッダー値の取得

**getHeader** 関数を使用して、特定のヘッダーの値を取得します。

例えば、次の式を使用すると、特定の出力が得られます。

式:

**request:getHeader("accept")**

出力:

text/xml



## ヘッダー名リストの取得

getHeaderNames 関数を使用して、一連のヘッダー名を値なしで取得します。

例えば、次の式を使用すると、特定の出力が得られます。

式:

```
request.getHeaderNames()
```

出力:

```
accept-language, client-ip, accept, postman-token, CustomHeader, ExecutionDate
```

## ヘッダー値またはデフォルトの取得

getHeaderOrDefault を使用して、ヘッダーが存在する場合はヘッダー値を取得し、ヘッダーが存在しない場合は何らかのデフォルト値を取得します。

この関数を使用するには、ヘッダー名とデフォルト値を式パラメータとして指定する必要があります。

デフォルト値は Not Available にします。

例えば、次の式を使用すると、有効な出力が得られます。

式:

```
concat ("The header I want is",  
        request.getHeaderOrDefault("postman-token", "Not Available" )
```

出力:

```
The header I want is 0023-a5dfd-8vdg3-n2b3
```

次の式を使用すると、Not Available の出力が得られます。

式:

```
concat ("The header I want is",  
        request.getHeaderOrDefault("special-token", "Not Available" )
```

出力:

```
The header I want is Not Available
```

これは、要求に、名前が special-token のヘッダーが含まれないためです。

## 特定のヘッダーが存在するかどうかの確認

headerExists 関数を使用して、特定の名前を持つヘッダーが存在するかどうかを確認します。

出力は、True または False のいずれかになります。

例えば、次の式を使用すると、特定の出力が得られます。

式:

```
request.headerExists("CustomHeader")
```

出力:

```
True
```

## メッセージイベントのヘッダー

プロセスにメッセージイベントがある場合、任意のヘッダー関数を使用して、メッセージイベントのヘッダーを取得できます。これを行うには、式パラメータとして、メッセージイベント名を大文字と小文字を区別して追加する必要があります。

例えば、名前が CustomHeader のヘッダーをメッセージイベント受信ステップに渡す場合、次の式を使用して、CustomHeader の値をメッセージイベント ME1 から取得します。

```
request.getHeader("CustomHeader", "ME1")
```

## HTTP 動詞の関数

要求で使用される HTTP 動詞とリソースパスセグメントを取得するには、HTTP 動詞の関数を使用します。HTTP 動詞の関数は、式エディタの **【要求コンテキスト】** セクションで利用できます。

次の HTTP 動詞の関数を使用できます。

### getRequestHTTPVerb

要求で使用されている HTTP 動詞を判別するには、getRequestHTTPVerb 関数を使用します。この関数によって、要求から HTTP 動詞が取得され、次の応答のいずれかが文字列形式で返されます。

- GET
- POST
- PATCH
- PUT
- DELETE

### getResourcePathSegments

REST 要求のすべてのまたは特定のリソースパスセグメントを取得するには、getResourcePathSegments 関数を使用します。この関数によって、トークンの文字列として値が返されます。

例えば、次の要求 URL があるとします。

```
https://na1.ai.dm-us.informaticacloud.com/active-bpel/rt/InitiateOrder/Orders/OrderID_100/quantity/5
```

request.getResourcePathSegments() 式を使用すると、次の応答が返されます。

```
[Orders, OrderID_100, quantity, 5]
```

特定のリソースパスセグメントを取得するには、数値修飾子を使用して、プロセス名からのリソースパスセグメントの位置を示します。

例えば、OrderID\_100 セグメントだけを応答で取得するには、次の式を使用します。

```
request.getResourcePathSegments()[2]
```

**注:** getResourcePathSegments 関数を使用して、SOAP 要求およびメッセージイベントのリソースパスセグメントを取得することはできません。getResourcePathSegments 関数を使用して、ホストコンテキストを取得することはできません。

## ヒューマンタスク XQuery ユーティリティ

プロセス内でヒューマンタスクステップを使用する場合、Process Designer で、または式を使用して値を設定するフィールドで、XQuery を操作するときにヒューマンタスクユーティリティ関数を使用できます。

ヒューマンタスクユーティリティ関数は、式エディタの **【ヒューマンタスクユーティリティ】** の下にあります。

次の表に、ヒューマンタスクユーティリティ関数を示します。

機能	構文	説明
getHumanTaskId	<code>task:getHumanTaskId(humanTaskStepName)</code>	<p>指定されたヒューマンタスクステップ名に関連付けられたヒューマンタスクの ID を返します。</p> <p>この ID は、ヒューマンタスクに関する情報の取得、タスクの再割り当て、タスクのキャンセルなどの他のアクションに使用できます。</p>
getHumanTaskInfo	<code>task:getHumanTaskInfo(humanTaskId)</code>	<p>指定されたヒューマンタスク ID に基づいて、ヒューマンタスクに関する次の情報を返します。</p> <ul style="list-style-type: none"> <li>- タスク ID</li> <li>- タスク名</li> <li>- ステータス</li> <li>- 優先度</li> <li>- タスクイニシエータ</li> <li>- 関係者</li> <li>- 潜在的な所有者</li> <li>- 管理者</li> <li>- タスク所有者</li> <li>- タスク作成日</li> <li>- タスクを作成したユーザー。</li> <li>- タスクアクティブ化時刻</li> <li>- タスク有効期限</li> <li>- スキップ可能オプション値</li> </ul>
reassignHumanTask	<code>task:reassignHumanTask(humanTaskId, users, groups)</code>	<p>指定されたユーザーおよびグループにヒューマンタスクを再割り当てし、タスク ID を返します。</p> <p>タスクを 1 人のユーザーに再割り当てすると、そのユーザーがタスクの所有者になります。タスクをグループまたは複数のユーザーに再割り当てする場合、ユーザーはタスクを引き受けてタスク所有者になる必要があります。</p> <p>複数のユーザーまたはグループを指定するには、カンマ文字を区切り文字として使用します。例えば、以下の構文を使用します。</p> <pre>task:reassignHumanTask(humanTaskId, ('user1', 'user2'), ('group1', 'group2'))</pre> <p>タスクをグループではなく特定のユーザーに再割り当てする場合は、次の構文を使用します。</p> <pre>task:reassignHumanTask(humanTaskId, 'user1',")</pre>
cancelHumanTask	<code>task:cancelHumanTask(humanTaskId)</code>	<p>ヒューマンタスクをキャンセルし、タスクを廃止します。タスク ID を返します。ヒューマンタスクがキャンセルされた後、プロセスは次のステップに進みます。</p>

# 単純型のリストを使用したフィールドの作成

プロセスおよびガイドでは、入力フィールド、出力フィールド、一時フィールド、メッセージに次のようなデータ型を使用できます。

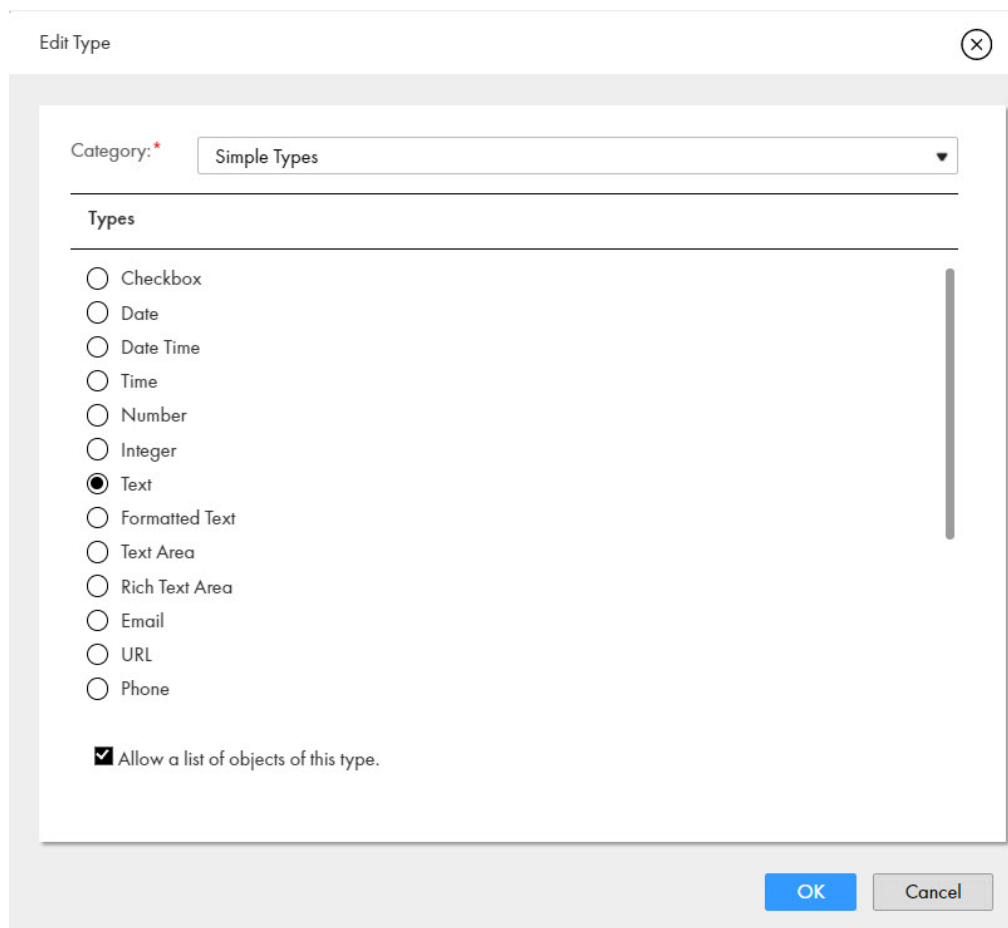
- チェックボックスリスト
- 日付リスト
- 日時リスト
- 時刻リスト
- 数値リスト
- 整数リスト
- テキストリスト

プロセスまたはガイドを作成するときは、カンマ区切りの値リストを定義できます。

**注:** メッセージはプロセスにのみ適用されます。

単純型のリストを使用したフィールドを作成するには、次の手順を実行します。

1. **【参照】** ページで、単純型のリストを使用したフィールドを作成するプロセスまたはガイドに移動します。
2. プロセスまたはガイドにステップを追加します。
3. **【開始】** ステップをクリックします。
4. 必要に応じて、**【入力フィールド】**、**【出力フィールド】**、**【一時フィールド】**、または **【メッセージ】** タブをクリックします。
5. **【追加】** アイコンをクリックします。
6. **【タイプ】** カラムで、**【タイプをさらに表示】** を選択します。  
**【タイプの編集】** ダイアログボックスが表示されます。  
次の図に、**【タイプの編集】** ダイアログボックスを示します。



7. **【カテゴリ】** リストで **【単純型】** を選択します。  
データ型のリストが **【タイプ】** セクションに表示されます。
8. **【チェックボックス】**、**【日付】**、**【日付/時刻】**、**【時間】**、**【数値型】**、**【整数型】**、または **【テキスト】** タイプを必要に応じて選択し、**【このタイプのオブジェクトのリストを許可します】** を選択します。
9. **【OK】** をクリックします。  
選択したタイプに基づいて、フィールドには対応する単純型のリストが設定されます。

## 単純型のリストを使用したプロセスの呼び出し

プロセスを呼び出すとき、値のリストを指定する形式は、HTTP 動詞のタイプとペイロード形式に応じて異なります。チェックボックス、日付、日付/時刻、時間、数値、整数、およびテキストの値のリストを指定できません。

GET 要求では、チェックボックス、日付、日付/時刻、時間、数値、整数、およびテキストの値のリストをアンパサンド (&) で区切って URL に指定します。

以下に例を示します。

https://na1.ai.dm-us.informaticacloud.com/active-bpel/public/rt/3TbGuTUy0cchqLy0IXoXsn/OrderManagementProcess?taskIds=TASK-123&taskIds=TASK-456&itemcount=5&itemcount=6&listdatetime=2022-08-24T21:32:52&listboolean=true&listboolean=false

POST 要求では、JSON を使用する場合、カンマ区切りの値の配列を指定します。

例えば、JSON には次のテキストを使用します。

```
{
  "taskIds": ["TASK-123", "TASK-456", "TASK-789" ]
  "itemcount": [5,6]
  "listdatetime": ["2022-08-24T21:32:52", "2022-08-25T22:32:52"]
  "listboolean": [true,false]
}
```

POST 要求では、XML を使用する場合、XML 形式でテキストを指定します。

例えば、XML には次のテキストを使用します。

```
<taskIds>TASK-123</taskIds>
<taskIds>TASK-456</taskIds>
<itemcount>5</itemcount>
<itemcount>6</itemcount>
<listdatetime>2022-08-24T21:32:52</listdatetime>
<listdatetime>2022-08-25T22:32:52</listdatetime>
<listboolean>true</listboolean>
```

SOAP 要求では、POST 要求と同様に XML 形式で値を指定します。

## カスタムタイプを使用したフィールドの作成

プロセスオブジェクトを使用して、カスタムタイプを使用したフィールドを作成できます。プロセスオブジェクトタイプのリストを作成することもできます。プロセスまたはガイドを作成するときは、入力フィールド、出力フィールド、一時フィールド、およびメッセージにプロセスオブジェクトデータ型を使用できます。

**注:** メッセージはプロセスにのみ適用されます。

カスタムタイプを使用したフィールドを作成するには、次の手順を実行します。

1. **【参照】** ページで、カスタムタイプを使用したフィールドを作成するプロセスまたはガイドに移動します。
2. プロセスまたはガイドにステップを追加します。
3. **【開始】** ステップをクリックします。
4. 必要に応じて、**【入力フィールド】**、**【出力フィールド】**、**【一時フィールド】**、または **【メッセージ】** タブをクリックします。
5. **【追加】** アイコンをクリックします。
6. **【タイプ】** カラムで、**【タイプをさらに表示】** を選択します。  
**【タイプの編集】** ダイアログボックスが表示されます。
7. **【カテゴリ】** リストで **【カスタムタイプ】** を選択します。

プロセスオブジェクトのリストが **【タイプ】** セクションに表示されます。  
次の図は、使用可能なプロセスオブジェクトのリストを示しています。

Category: \* Custom Types

Types

- Alert Details
- Any
- Data Marketplace Parameters
- Data Marketplace Path Variables
- Data Marketplace Query Parameters
- ProcessWithListOfTextRequest
- Task Action
- Task Actions
- Task Type
- Task Types

Allow a list of objects of this type.

OK Cancel

8. 必要に応じてプロセスオブジェクトを選択します。プロセスオブジェクトのリストを使用するには、**[このタイプのオブジェクトのリストを許可します]** を選択します。
9. **[OK]** をクリックします。

選択したプロセスオブジェクトに基づいて、フィールドにはプロセスオブジェクトタイプまたはプロセスオブジェクトタイプのリストが設定されます。

## 定義された接続タイプを使用したフィールドの作成

アプリケーション接続に関連付けられたオブジェクトを使用して、定義された接続タイプを使用したフィールドを作成できます。オブジェクトタイプのリストを作成することもできます。プロセスまたはガイドを作成するときは、入力フィールド、出力フィールド、一時フィールド、およびメッセージにオブジェクトデータ型を使用できます。

**注:** メッセージはプロセスにのみ適用されます。

定義された接続タイプを使用したフィールドを作成するには、次の手順を実行します。

1. **[参照]** ページで、定義された接続タイプを使用したフィールドを作成するプロセスまたはガイドに移動します。
2. プロセスまたはガイドにステップを追加します。
3. **[開始]** ステップをクリックします。
4. 必要に応じて、**[入力フィールド]**、**[出力フィールド]**、**[一時フィールド]**、または **[メッセージ]** タブをクリックします。

5. **【追加】** アイコンをクリックします。
6. **【タイプ】** カラムで、**【タイプをさらに表示】** を選択します。  
**【タイプの編集】** ダイアログボックスが表示されます。
7. **【カテゴリ】** リストで **【定義された接続タイプ】** を選択します。
8. **【接続】** リストで、接続を選択します。

接続で使用可能なオブジェクトのリストが **【タイプ】** セクションに表示されます。  
次の図は、接続に関連付けられたオブジェクトを示しています。

Category: \* Connection defined Types

Connection: Connections > Email > ExchangeWebServices

The Field can be one of several types.

Types

Item

Recipient

Allow a list of objects of this type.

OK Cancel

9. 必要なオブジェクトを選択します。オブジェクトのリストを使用するには、**【このタイプのオブジェクトのリストを許可します】** を選択します。
10. **【OK】** をクリックします。

選択したオブジェクトに基づいて、フィールドにはオブジェクトタイプまたはオブジェクトタイプのリストが設定されます。

**注:** **【フィールドはいくつかのタイプのうちのいずれかになります】** オプションは、将来使用するために予約されています。



## 第 2 章

# プロセスのデザイン

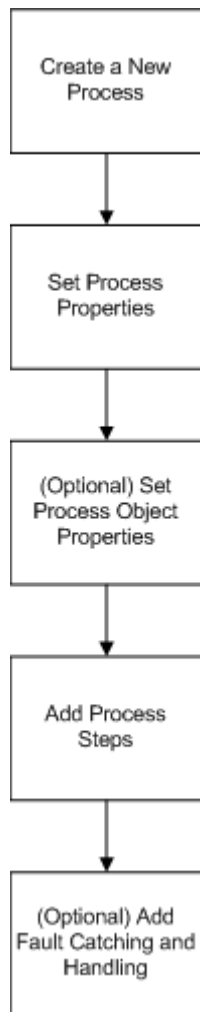
Process Designer を使用してアプリケーションを統合すると、次のことが可能になります。

- データの決定、サブプロセス、並列パス、受信、待機など、各種ステップタイプからプロセスを構築します。
- プロセス内のステップを自動的にリンクするには、データを選択し、アクション、バインディング、その他のプロセスプロパティを定義します。
- プロセスの設計中に、プロセスと必要な接続をテストします。
- プロセスが自動的にデプロイされ、REST/XML または JSON サービスとして呼び出すことができるように、プロセスをパブリッシュします。

例えば、Salesforce の注文処理を合理化するプロセスは、次のように定義できます。

- 配送情報を取得するためにサービスを呼び出します。
- オンプレミスに存在する在庫データベースから、現在の価格設定を検索します。
- 注文詳細を SAP に送信します。

次の図は、プロセスを設計する方法を示しています。



## プロセスの作成

1. アプリケーションの統合で、**【新規】** を選択します。
2. **【新しいアセット】** ダイアログボックスで、**【プロセス】** > **【プロセス】** の順に選択し、**【作成】** をクリックします。

## プロセスプロパティの設定

1. アプリケーションの統合で、**【新規】** をクリックして **【新しいアセット】** ウィンドウを開きます。
2. **【プロセス】** > **【プロセス】** を選択して、**【作成】** をクリックします。  
Process Designer が開きます。

次の図は、【プロセスのプロパティ】パネルを示しています。

Process 1 Properties

**General**

Start

Input Fields

Output Fields

Temp Fields

Messages

Advanced

Notes

Step Type: Start

Name: \* Process 1

API Name:   Override API Name

Location: Default

Description:

3. 必須プロパティを入力します。
4. 必要に応じて、プロセスプロパティを表示または編集するには、【開始】ステップをクリックします。
5. 【保存】をクリックします。

## 全般プロパティ

プロセスの次の全般プロパティを指定できます。

プロパティ	説明
名前	<p>必須。Process Designer でプロセスを識別するためのわかりやすい名前であり、サブプロセスステップなどの他のオブジェクトでプロセスを使用できる場合に表示される名前。</p> <p>名前にはマルチバイト文字を含めることができ、80 文字以内にする必要があります。</p> <p>パブリッシュされたプロセスの名前を変更するには、まずプロセスのパブリッシュを解除する必要があります。次に、名前を変更し、プロセスを再パブリッシュします。</p>
API 名のオーバーライド	<p>オプション。指定した名前のプロセスをパブリッシュするときに自動生成される API 名をオーバーライドします。このオプションを選択すると、【API 名】フィールドが使用できるようになります。</p>
API 名	<p>【API 名のオーバーライド】オプションを選択する場合は必須です。</p> <p>プロセスに対して自動生成された API 名をオーバーライドする一意の API 名。このフィールドで指定する API 名は、生成されたサービス URL で使用されます。</p> <p>API 名に使用できるのは英数字、アンダースコア ( _ )、ハイフン ( - ) のみです。80 文字以内にする必要があります。</p> <p>パブリッシュされたプロセスの API 名を変更するには、まずプロセスのパブリッシュを解除する必要があります。次に、API 名を変更し、プロセスを再パブリッシュします。</p> <p>API 名をオーバーライドし、プロセスをインポートすると、インポートされたプロセスでは指定した API 名が使用されます。ただし、指定したものと同一 API 名の既存プロセスがある場合、アプリケーション統合では重複を避けるためにインポートされたプロセスに対し自動生成された API 名が使用されます。また、アプリケーション統合では【API 名のオーバーライド】フィールドが無効化されます。同様に、プロセスをコピーすると、アプリケーション統合では重複を避けるためにコピーされたプロセスに対し自動生成された API 名が使用され、また、【API 名のオーバーライド】フィールドが無効化されます。</p> <p>API 名をオーバーライドし、プロセスを別の場所に移動すると、アプリケーション統合では指定した API 名が保持されます。</p>

プロパティ	説明
場所	オプション。プロセスを配置するプロジェクトまたはフォルダの場所。プロセスの場所を選択するには、該当するプロジェクトまたはフォルダを参照するか、デフォルトの場所を使用します。
説明	オプション。プロセスの説明。

## 開始プロパティ

**【開始】** タブをクリックして、バインディングタイプ、アクセスの詳細、およびランタイム環境の詳細を定義します。

プロセス用に次の開始プロパティを定義できます。

### バインディング

**【バインディング】** プロパティでは、プロセスの呼び出しおよび実行方法を定義します。

次の値から選択することができます。

- **REST/SOAP:** サービス URL を使用してプロセスを実行する場合は、**【REST/SOAP】** バインディングタイプを選択します。このオプションを選択した場合は、**【許可されたグループ】** および **【許可されたユーザー】** フィールドを使用して、パブリッシュされたプロセスを API として実行できるグループおよびユーザーを定義できます。ユーザーを割り当てない場合は、**【匿名アクセスを許可】** オプションを選択します。
- **イベント:** 指定したイベントが発生したときにプロセスが呼び出されるようにするには、**【イベント】** バインディングタイプを選択します。例えば、ファイルシステムへのファイル到着などのイベントが発生したときにプロセスを呼び出すことができます。イベントソースを選択できる **【イベントソース名】** フィールドを使用できます。

デフォルトは **【REST/SOAP】** です。

### 許可されたグループ

実行時にプロセスのサービス URL へのアクセス権を持つグループ。グループ内のユーザーが実行の特権も持っている場合は、プロセスのサービス URL を起動できます。

ユーザーグループにサービス URL へのアクセス権を付与する場合は、**【許可されたグループ】** フィールドを使用します。例えば、「Order Approvers」というグループがある場合を考えます。**【許可されたグループ】** フィールドに「**Order Approvers**」と入力すると、このグループ内のすべてのユーザーがサービス URL にアクセスできるようになります。

次の図は、**【許可されたグループ】** フィールドに Order Approvers グループが追加されているところを示します。

Allowed Groups:

Allowed Users:

フィールドには、複数のグループを入力できます。

**【許可されたグループ】** フィールドと **【許可されたユーザー】** フィールドを設定できます。ユーザーが **【許可されたグループ】** または **【許可されたユーザー】** のいずれかのカテゴリに当てはまる場合、そのユーザーはサービス URL にアクセスできます。

**【許可されたグループ】** フィールドまたは **【許可されたユーザー】** フィールドを設定し、**【匿名アクセスを許可】** オプションを選択すると、入力したグループまたはユーザーの詳細情報は表示されなくなります。

**【匿名アクセスを許可】** オプションを選択すると、**【許可されたグループ】** フィールドおよび **【許可されたユーザー】** フィールドは編集できなくなります。

**【許可されたグループ】** フィールドまたは **【許可されたユーザー】** フィールドを設定しない場合、または **【匿名アクセスを許可】** オプションを選択しない場合、このプロセスで利用できるサービス URL はありません。ただし、プロセスをサブプロセスとして呼び出すことはできます。

API マネージャにプロセスを表示する場合は、**【匿名アクセスを許可】** オプションを選択するか、**【許可されたグループ】** フィールドまたは **【許可されたユーザー】** フィールドを設定する必要があります。API 管理の詳細については、「API マネージャ」のヘルプを参照してください。

プロセスが **【許可されたグループ】** または **【許可されたユーザー】** フィールドを使用している場合、次のいずれかの方法を使用して、Postman などの REST クライアントを通じてプロセスを呼び出すことができます。

- 基本認証を使用し、ユーザー名とパスワードを入力します。
- ユーザー名とパスワードを指定せずにプロセスを呼び出すには、HTTP ヘッダーのセッション ID を使用します。

例えば、Postman を使用してプロセスサービス URL を呼び出すには、次のいずれかの方法で GET 要求を認証します。

- 基本認証を使用し、Informatica Intelligent Cloud Services のユーザー名およびパスワードを指定します。

例:

```
GET <Informatica Intelligent Cloud Services URL>/active-bpel/rt/<API unique name>
Accept: application/json
Authorization: Basic Auth
username: <Informatica Intelligent Cloud Services user name>
password: <Informatica Intelligent Cloud Services password>
```

- HTTP ヘッダーの IDS-SESSION-ID を使用します。

例:

```
GET <Informatica Intelligent Cloud Services URL>/active-bpel/rt/<API unique name>
Accept: application/json
IDS-SESSION-ID: <sessionId>
```

SESSION-ID を取得するには、Platform REST API バージョン 3 ログインリソースを使用します。ログインリソースの詳細は、『*REST API リファレンス*』を参照してください。

### 許可されたユーザー

実行時にプロセスのサービス URL へのアクセス権を持つユーザー。ユーザーが実行の特権も持っている場合は、プロセスのサービス URL を起動できます。

特定のユーザーにサービス URL へのアクセス権を付与する場合は、**【許可されたユーザー】** フィールドを使用します。

次の図では、**【許可されたユーザー】** フィールドにユーザー jsmith が表示され、**【許可されたグループ】** フィールドに **【Order Approvers】** が表示されています。

The image shows a configuration interface with two input fields. The first field is labeled 'Allowed Groups:' and contains the text 'Order Approvers' followed by a small 'x' icon. The second field is labeled 'Allowed Users:' and contains the text 'jsmith' followed by a small 'x' icon.

このプロセスで、Order Approvers グループのユーザーと、ユーザー jsmith はサービス URL へのアクセス権を持ちます。

フィールドには、複数のユーザーを入力できます。

**【許可されたグループ】** フィールドまたは **【許可されたユーザー】** フィールドを設定し、**【匿名アクセスを許可】** オプションを選択すると、入力したグループまたはユーザーの詳細情報は表示されなくなります。

**【匿名アクセスを許可】** オプションを選択すると、**【許可されたグループ】** フィールドおよび **【許可されたユーザー】** フィールドは編集できなくなります。

**【許可されたグループ】** フィールドまたは **【許可されたユーザー】** フィールドを設定しない場合、または **【匿名アクセスを許可】** オプションを選択しない場合、このプロセスで利用できるサービス URL はありません。ただし、プロセスをサブプロセスとして呼び出すことはできます。

API マネージャにプロセスを表示する場合は、**【匿名アクセスを許可】** オプションを選択するか、**【許可されたグループ】** フィールドまたは **【許可されたユーザー】** フィールドを設定する必要があります。API 管理の詳細については、「API マネージャ」のヘルプを参照してください。

プロセスが **【許可されたグループ】** または **【許可されたユーザー】** フィールドを使用している場合、次のいずれかの方法を使用して、Postman などの REST クライアントを通じてプロセスを呼び出すことができます。

- 基本認証を使用し、ユーザー名とパスワードを入力します。
- ユーザー名とパスワードを指定せずにプロセスを呼び出すには、HTTP ヘッダーのセッション ID を使用します。

例えば、Postman を使用してプロセスサービス URL を呼び出すには、次のいずれかの方法で GET 要求を認証します。

- 基本認証を使用し、Informatica Intelligent Cloud Services のユーザー名およびパスワードを指定します。

例:

```
GET <Informatica Intelligent Cloud Services URL>/active-bpel/rt/<API unique name>  
Accept: application/json  
Authorization: Basic Auth  
username: <Informatica Intelligent Cloud Services user name>  
password: <Informatica Intelligent Cloud Services password>
```

- HTTP ヘッダーの IDS-SESSION-ID を使用します。

例:

```
GET <Informatica Intelligent Cloud Services URL>/active-bpel/rt/<API unique name>  
Accept: application/json  
IDS-SESSION-ID: <sessionId>
```

SESSION-ID を取得するには、Platform REST API バージョン 3 ログインリソースを使用します。ログインリソースの詳細は、『*REST API リファレンス*』を参照してください。

#### API ゲートウェイからの HTTP 認証要求のみを承認します。

このオプションを使用して、プロセスに対し OAuth2 認証を有効にし、API を API ゲートウェイからの HTTP 認証要求のみを承認するように設定します。

API を API ゲートウェイから呼び出すには、次のユーザーが設定タスクを完了する必要があります。

#### API プロバイダ

API プロバイダは次のタスクを実行する必要があります。

1. API マネージャでプロセスに対して管理対象 API を作成し、管理対象 API に対して OAuth2 認証を有効にします。
2. API マネージャで OAuth2 クライアントを作成し、クライアント ID とクライアントシークレットを生成します。

3. Informatica Intelligent Cloud Services OAuth 2.0 サーバー URL とクライアント資格情報を API コンシューマに送信します。

#### API コンシューマ

API コンシューマは次のタスクを実行する必要があります。

1. Informatica Intelligent Cloud Services OAuth 2.0 サーバーに対して認証し、OAuth 2.0 クライアント資格情報を使用して、OAuth 2.0 認証トークンを生成します。
2. OAuth 2.0 認証トークンを使用して API を呼び出します。

OAuth2 クライアントの作成と OAuth2 認証を使用した API の呼び出しに関する詳細については、「API マネージャ」のヘルプを参照してください。

**注:** クラウドサーバーで実行するプロセスにのみ **[API ゲートウェイからの HTTP 認証要求のみを承認します]** オプションを使用できます。**[API ゲートウェイからの HTTP 認証要求のみを承認します]** オプションを使用しているプロセスは、1 つまたは複数のプロセス入力を使用してスケジュールし、実行することもできます。

#### 匿名アクセスを許可

選択すると、誰でも Process Designer でこのプロセスを使用できるようになります。これは、このプロセスへのアクセスが、特定のユーザーやユーザーグループに縛られないことを意味します。

**[匿名アクセスを許可]** オプションを選択すると、**[許可されたグループ]** フィールドおよび **[許可されたユーザー]** フィールドは編集できなくなります。

**[許可されたグループ]** フィールドまたは **[許可されたユーザー]** フィールドを設定し、**[匿名アクセスを許可]** オプションを選択すると、入力したグループまたはユーザーの詳細情報は表示されなくなります。

**[許可されたグループ]** フィールドまたは **[許可されたユーザー]** フィールドを設定しない場合、または **[匿名アクセスを許可]** オプションを選択しない場合、このプロセスで利用できるサービス URL はありません。ただし、プロセスをサブプロセスとして呼び出すことはできます。

API マネージャにプロセスを表示する場合は、**[匿名アクセスを許可]** オプションを選択するか、**[許可されたグループ]** または **[許可されたユーザー]** フィールドを使用する必要があります。API 管理の詳細については、「API マネージャ」のヘルプを参照してください。

#### 適用対象

プロセスに関連付けられているオブジェクトのタイプ。このリストに表示されるオブジェクトは、個別に定義したプロセスオブジェクトまたは定義済みの接続内で生成したプロセスオブジェクトです。リスト内をドリルダウンしてプロセスオブジェクトと接続を表示し、これらの項目の 1 つを選択します。プロセスが別のプロセスに組み込まれる場合にのみ、プロセスオブジェクトを選択します。

**注:** パブリッシュされたプロセスの **[適用対象]** プロパティを変更するには、まずプロセスのパブリッシュを解除する必要があります。次に、**[適用対象]** プロパティを変更し、プロセスを再パブリッシュします。

次の 2 つの特別な値があります。

- **任意:** プロセスを特定のオブジェクトに関連付ける必要がない場合は、**[任意]** を選択します。このオプションを選択すると、プロセスが、オブジェクトのフィールドに自動的にアクセス権を持つことはありません。プロセスが新しいオブジェクトを作成し、既存のオブジェクト内の情報にアクセスしない場合は、このオブジェクトタイプを使用します。
- **ホーム (Salesforce) :** オブジェクトタイプにかかわらず、プロセスがホームから、またはユーザーがすべてのプロセスを起動した任意の初期の場所から表示されるようにするには、**[ホーム]** を選択します。

**注:** プロセスを直接呼び出す場合は、[適用対象] の設定は【任意】またはサービス内の特定のオブジェクト（プロセスオブジェクト以外）のいずれかである必要があります。

### 実行日時

このリストを使用して、このプロセスを実行する必要がある場所を指定します。選択できるのは、【クラウドサーバー】、特定のエージェント、または Secure Agent グループです。

プロセスがイベントベースのコネクタを使用する場合は、コネクタの実行場所と同じエージェントでプロセスを実行するようにします。

プロセスがパブリッシュ済み状態の場合、【実行日時】 リストは無効になっています。【実行日時】 リストを有効にするには、まずプロセスのパブリッシュを解除する必要があります。次に、【実行日時】 フィールド値を変更し、プロセスを再パブリッシュします。

プロセスにヒューマンタスクステップを含める場合は、クラウドサーバーでプロセスを実行する必要があります。

### ユーザーを指定して実行（Salesforce のみ）

Salesforce の場合のみ、プロセスを【現在のユーザー】または【システム】として実行するかどうかを選択できます。

プロセスを実行しているユーザーと同じ権限が必要な場合、【現在のユーザー】を選択します。

プロセスにシステムレベルのアクセス権限がないと、プロセスのユーザーがアクセスできない可能性があるアクションを実行できるようにするために、プロセスにシステムレベルのアクセス権限が必要な場合、【システム】を選択します。例えば、ユーザーにアカウントオブジェクトへのアクセス権限がなくても、プロセスが【システム】として実行されていると、プロセスはプロセスを実行するために必要な範囲でアカウントオブジェクトにアクセスできます。

## 入力フィールドのプロパティ

プロセスに対して使用できる入力フィールドは、プロセスの適用先のオブジェクトに含まれるフィールドです。これは、[適用対象] プロパティ（【開始】 タブ）で指定されます。

[入力] フィールドの値は、プロセスのすべてのステップ内で使用できます。プロセスで使用するために、入力フィールドのカテゴリに対して名前とタイプを定義する必要があります。

プロパティ	説明
入力形式	入力フィールドが 1 つまたは複数のフィールドを表せるか、要求のコンテンツ全体を表せるかを決定します。【ペイロード全体】を選択した場合、入力フィールドは、要求のコンテンツ全体を表します。例えば、顧客のプロセスオブジェクトがある場合、そのコンテンツは次のようになる場合があります。 <pre>"customer": {"name": "Joe Smith", "street": "3 Main St", "city": "Shelton", "state": "CT"}</pre> このオプションを選択した場合、次の内容がポストされます。 <pre>{"name": "Joe Smith", "street": "3 Main St", "city": "Shelton", "state": "CT"}</pre>
名前	入力フィールドの名前です。



プロパティ	説明
タイプ	入力フィールドのタイプ。例えば、[日付]、[日付/時刻]、[時間]、[整数]、または [テキスト] を選択します。 単純型、カスタムタイプ、または定義された接続タイプの入力フィールドを追加することもできます。入力フィールドを追加するには、[タイプをさらに表示] を選択し、[タイプの編集] ダイアログボックスで、[カテゴリ] に [単純型]、[カスタムタイプ]、または [定義された接続タイプ] を選択します。
説明	入力フィールドの説明です。
必須	フィールドがプロセスの実行に必要な場合は、[必須] を選択します。

フィールドの詳細については、「[データ型とフィールドプロパティの概要](#)」(ページ 10)を参照してください。

## XML ペイロード

JSON 構文では、ルートコンテナは必要ありません。ただし、ペイロードが XML の場合、ルート要素は必要です。XML を送信する場合、Process Designer は常にルートノードを追加します。ペイロードが XML の場合、Process Designer は、要求を送信するときに、常にルートノードを追加します。

例えば、XML がラップされている場合は、次のようになります。

```
<root>
  <oAddressP0>
    <Street>3 Enterprise Drive</Street>
    <State>CT</State>
    <Zipcode>6484.0</Zipcode>
    <City>Shelton</City>
  </oAddressP0>
  <oAddressP0>
    <Street>31 Enterprise Drive</Street>
    <State>CT1</State>
    <Zipcode>106484.0</Zipcode>
    <City>Shelton1</City>
  </oAddressP0>
</root>
```

XML がラップされていない場合は、次のようになります。

```
<root>
  <_1>
    <Street>3 Enterprise Drive</Street>
    <State>CT</State>
    <Zipcode>6484.0</Zipcode>
    <City>Shelton</City>
  </_1>
  <_2>
    <Street>31 Enterprise Drive</Street>
    <State>CT1</State>
    <Zipcode>106484.0</Zipcode>
    <City>Shelton1</City>
  </_2>
</root>
```

## 出力フィールドのプロパティ

プロセスに対して使用できる出力フィールドは、プロセスの適用先のオブジェクトに含まれるフィールドです。これは、[適用対象] プロパティ ([開始] タブ) で指定されます。

[出力] フィールドの値は、プロセスの実行時に設定され、後続のステップ内で使用できるようになります。出力フィールドは、組み込みプロセス内で使用する目的でのみ、定義します。これらの出力フィールドは、値が

組み込みプロセスから返されるまでは、リストに表示されません。プロセスで使用するために、出力フィールドのカテゴリに対して名前とタイプを定義します。

プロパティ	説明
出力形式	出力フィールドが1つまたは複数の応答フィールドを表せるか、応答のコンテンツ全体を表せるかを決定します。[ペイロード全体]を選択した場合、出力フィールドは、応答のコンテンツ全体を表します。
名前	出力フィールドの名前です。
タイプ	出力フィールドのタイプです。例えば、[日付]、[日付/時刻]、[時間]、[整数]、または[テキスト]を選択します。 単純型、カスタムタイプ、または定義された接続タイプの出力フィールドを追加することもできます。入力フィールドを追加するには、[タイプをさらに表示]を選択し、[タイプの編集]ダイアログボックスで、[カテゴリ]に[単純型]、[カスタムタイプ]、または[定義された接続タイプ]を選択します。
説明	出力フィールドの説明です。
初期値	出力フィールドの初期値です。

フィールドの詳細については、「[データ型とフィールドプロパティの概要](#)」(ページ 10)を参照してください。

## 一時フィールドのプロパティ

プロセスに対して使用できる一時フィールドは、プロセスの適用先のオブジェクトに含まれるフィールドです。これは、[適用対象] プロパティ ([開始] タブ) で指定されます。

一時フィールドの値は、現在のプロセスのすべてのステップ内で使用されます。ただし、一時フィールドの値は、組み込みプロセス内で使用できるようにはなりません。プロセスで使用するために、一時フィールドのカテゴリに対して名前とタイプを定義します。

プロパティ	説明
名前	一時フィールドの名前です。
タイプ	一時フィールドのタイプです。例えば、[日付]、[日付/時刻]、[数値]、[パーセント]、[時間]、[通貨]、[整数]、または[テキスト]を選択します。 単純型、カスタムタイプ、または定義された接続タイプの一時フィールドを追加することもできます。入力フィールドを追加するには、[タイプをさらに表示]を選択し、[タイプの編集]ダイアログボックスで、[カテゴリ]に[単純型]、[カスタムタイプ]、または[定義された接続タイプ]を選択します。
説明	一時フィールドの説明です。
初期値	一時フィールドの初期値です。データ型が数値、パーセント、通貨、または整数の場合、デフォルトは0です。

フィールドの詳細については、「[データ型とフィールドプロパティの概要](#)」(ページ 10)を参照してください。

## メッセージのプロパティ

必要に応じて、このプロセスに1つまたは複数のメッセージイベントを定義できます。

各メッセージに対し、次のプロパティ（開始イベントと類似）を定義できます。

プロパティ	説明
入力フィールド	フィールド名を指定します。 各入力フィールドのタイプを選択します。例えば、[日付]、[日付/時刻]、[時間]、[整数]、または[テキスト]を選択します。 単純型、カスタムタイプ、または定義された接続タイプの入力フィールドを追加することもできます。入力フィールドを追加するには、[タイプをさらに表示]を選択し、[タイプの編集]ダイアログボックスで、[カテゴリ]に[単純型]、[カスタムタイプ]、または[定義された接続タイプ]を選択します。 メッセージが有効であるためにこのフィールドが必要な場合、[必須]を選択します。 このフィールドが、関連メッセージ受信イベントに使用される場合、[関連の使用]を選択します。
出力フィールド	フィールド名を指定します。 各出力フィールドのタイプを選択します。例えば、[日付]、[日付/時刻]、[時間]、[整数]、または[テキスト]を選択します。 単純型、カスタムタイプ、または定義された接続タイプの出力フィールドを追加することもできます。入力フィールドを追加するには、[タイプをさらに表示]を選択し、[タイプの編集]ダイアログボックスで、[カテゴリ]に[単純型]、[カスタムタイプ]、または[定義された接続タイプ]を選択します。
バインディング	メッセージの呼び出し方法に応じて、[REST/SOAP] または [イベント] を選択します。
許可ユーザー/グループ	このメッセージにアクセスできるユーザーおよびグループを入力します。
匿名アクセスを許可	選択した場合、どのユーザーも認証を受けずにこのメッセージを使用できます。 注: このオプションを有効にすると、Process Designer は [許可ユーザー/グループ] を使用して設定された制限を無視します。
入力形式がペイロード全体	入力フィールドは、このメッセージに含まれた要求のコンテンツ全体を表します。
出力形式がペイロード全体	出力フィールドは、応答のコンテンツ全体を表します。これにより、ペイロードが変わる方法は、入力フィールドの場合と同じです。

フィールドの詳細については、「[データ型とフィールドプロパティの概要](#)」（ページ 10）を参照してください。

### 関連メッセージイベント

プロセスを設計するときに、受信ステップに関連してメッセージイベントを使用し、プロセスの実行が開始された後に、プロセスと対話できます。例えば、次のようにできます。

- プロセスのステータスのクエリ
- データの更新、またはプロセスへのコントロールの提供

プロセスエンジンは、プロセスを一意に識別する1つまたは複数の入力パラメータに基づいて、関連メッセージイベントを対象のビジネスプロセスインスタンスとマッチングします。

例えば、注文発注プロセスに OrderId プロパティがあるとします。メッセージは OrderId の値を注文プロセスのさまざまな部分に異なる名前で格納でき、特定のプロセスインスタンスを一意的 OrderId と関連付けます。相関する OrderId を持つメッセージが到着すると、正しいプロセスインスタンスにディスパッチされます。必要な数の相関セットを作成できます。

相関によって着信メッセージは対象のビジネスプロセスインスタンスとマッチングされるため、ビジネスプロセスで非同期の要求/応答パターンをサポートできます（これに対し、同期呼び出しの場合、メッセージコンテキストがスタック上または TCP 接続を介して維持されるため、相関に依存する必要がありません）。

メッセージイベントは、プロセスレベルで定義します。メッセージイベントは、中断または非中断のいずれかにできます。

## 相関を使用する前に

相関メッセージ受信イベントのセットを定義する前に、次の手順を実行します。

- 相関が必要な Process Designer の手順を特定します。共通のデータを 1 つ以上共有する必要があります。
- 共通データを識別するプロパティを定義します。
- Process Designer でこれを実装できるように、データの相関セットを定義します。

## メッセージイベントのデザインガイドライン

相関メッセージイベントを定義するときは、次の点に注意してください。

- プロセスプロパティで定義する各メッセージイベントについて、少なくとも 1 つの入力フィールドを【**相関の使用**】として指定する必要があります。この入力フィールドが参照タイプ（プロセスオブジェクト）の場合、単純型のフィールドへのパスも指定します（「フィールド」または「計算式」を選択）。
- イベントの入出力はプロセスのフィールドとして使用できますが、受信ステップのダウンストリームでのみ使用できます（サービスステップの出力と同様）。
- 複数のメッセージイベント定義を使用し、異なるメッセージイベント内の 2 つの入力パラメータの名前が同じ場合、入力パラメータを相関に使用するには、それらが同じタイプと同じ相関パスである必要があります。そうすることで、ステップの相関の値が 1 つであることを保証できます。

中断または非中断のどちらであるかも指定します。

中断メッセージイベントを使用した場合、次のようになります。

- イベントはステップを終了します。
- イベントパスは、メインのパスと統合できます。
- イベントパスにはマイルストーンが必要です。このマイルストーンは、メッセージイベント受信ステップへの応答として機能します。

非中断メッセージイベントを使用した場合、次のようになります。

- プロセスはメッセージイベントを待機しません。
- イベントパスは、終了ステップで終了する必要があります。これは、メッセージイベント受信ステップへの応答として機能します。

## メッセージイベントを呼び出すためのクライアントの相互作用

メッセージイベントはプロセスと同様に呼び出せますが、使用するエンドポイントが異なります。例えば、プロセスを開始するエンドポイントは次のようになります。

```
.../active-bpel/public/rt/00000I/OrderProcessor
```

特定のメッセージイベントを呼び出すには、次のようなエンドポイントを使用します。

```
.../active-bpel/public/rt/00000I/OrderProcessor/event/CheckOrderStatus
```

相関メッセージイベントに関連する概念の詳細については、プロセス開発者向けガイドも参照してください。

## 詳細プロパティ

詳細プロパティを設定して、プロセスに対するフォールトの処理、追跡、およびオリジン間リソース共有 (CORS) サポートを定義することができます。

以下の詳細プロパティを設定できます。

### フォールト時に一時停止

デフォルトでは、キャッチされないフォールトが発生した場合、プロセスサーバーによりプロセスが終了されます。キャッチされないフォールト時に個々のプロセスを中断するように、Process Designer を設定できます。

**[フォールト時に一時停止]** オプションを選択した場合、キャッチされないフォールトが発生すると、プロセスサーバーはプロセスを中断します。**[フォールト時に一時停止]** オプションを選択しない場合、フォールトが発生し、プロセスがそのフォールトを明示的に処理しないときには、プロセスサーバーはプロセスを終了します。

プロセスが中断状態にある場合は、エラー状態を確認して特定してください。その後、アクティビティを再試行するか完了できます。例えば、従業員または顧客のオンボーディング中に、エラー状態を捕捉できます。エラーをすべて解決すると、プロセスは中断状態から脱し、フォールトが発生したステップから実行を続けます。

**[フォールト時に一時停止]** オプションを選択すると、Process Designer は完全な永続性を使用します。

**注:** プロセス固有のプロパティは、アプリケーション統合コンソールでユーザーが指定するその他の例外管理設定と連動します。

フォールトの管理と修正方法については、『[監視](#)』を参照してください。

### トレースレベル

次のいずれかのトレースレベルを設定して、プロセスのランタイム時におけるロギングのタイプを決定することができます。

- なし
- 簡易
- ノーマル
- 詳細

Process Designer で設定するトレースレベルは、次の表に示すように、Process Developer の永続レベルとログレベルにマッピングされます。

トレースレベル	永続レベル	ログレベル
なし	簡易	なし
簡易	簡易	フォルト
ノーマル	簡易	サービスデータを使用した実行
詳細	最終段階	データを使用した実行

### 永続レベル

ここで設定するトレースレベルにかかわらず、プロセスサーバーは、次の状況では完全な永続性を使用します。

- **[キャッチされないフォールト時に中断]** オプションを選択した場合

- プロセスにタイマ（待機ステップなど）、メッセージイベント、または受信イベントが含まれている場合
- プロセスが、サブプロセス連携を提供するために別のプロセスを呼び出した場合

各種永続レベルの詳細については、『[監視](#)』を参照してください。

## ログレベル

プロセスサーバーのログレベルは、個々のプロセスに対して設定するログレベルよりも低い場合があります。プロセスレベルで設定するログレベルにかかわらず、プロセスサーバーに指定した最大プロセスログレベルをロギングが超えることはありません。トレースレベルの表は、ログレベルと永続レベルの間のマッピングを示しています。

各種ログレベルの詳細については、『[管理者](#)』のオンラインヘルプを参照してください。

## トレースレベルの自動リセット

クラウドサーバー上で実行するように構成されたプロセスで、トレースレベルを【[詳細](#)】または【[標準](#)】に設定している場合、トレースレベルは必要に応じて、サーバーによって【[フォールト](#)】にリセットされます。この自動リセット機能は、実行時ログによるプロセス実行時間を短縮するのに役立ちます。

**注:** トレースレベルの自動リセット機能はテクニカルプレビュー段階です。テクニカルプレビュー機能はサポートされていますが、保証対象外で本番環境には対応していません。非本番環境でのみ使用することをお勧めします。Informatica では、本番環境用に次の GA リリースでプレビュー機能を導入するつもりですが、市場や技術的な状況の変化に応じて導入しない場合もあります。詳細については、Informatica グローバルカスタマサポートにお問い合わせください。

## CORS の有効化

デフォルトでは、API コンシューマは、サービス URL の一部であるドメインからプロセスを呼び出すことができます。API コンシューマは、GET、POST、PATCH、PUT、DELETE の各動詞を使用してプロセスを呼び出すことができます。

**[CORS の有効化]** オプションを選択すると、プロセスに対するオリジン間リソース共有 (CORS) サポートが有効になります。CORS サポートで許可されるドメインおよび動詞を指定することもできます。

**[ドメイン]** フィールドには、API コンシューマがプロセスを呼び出すことができる追加の許可されているドメインを指定します。すべてのドメインで CORS を有効にするには、ワイルドカード文字 (\*) を入力します。特定のドメインで CORS を有効にするには、ドメインのカンマ区切りリストを入力します。例えば、abc.domain.com ドメインと xyz.domain.com ドメインで CORS を有効にするには、次の句を入力します。

abc.domain.com,xyz.domain.com

CORS を有効にしても **[ドメイン]** フィールドを空のままにすると、サービス URL の一部であるドメインからしかプロセスを呼び出すことはできません。

**[ドメイン]** フィールドで指定したドメインからプロセスを呼び出す場合に API コンシューマが使用できる、許可されている動詞を **[動詞]** フィールドに指定します。特定の動詞に対して CORS を有効にするには、動詞のカンマ区切りリストを入力します。例えば、POST、PUT、GET の各動詞で CORS を有効にするには、次の句を入力します。

POST,PUT,GET

デフォルトはすべての動詞です。

API コンシューマはサービス URL の一部であるドメインから、プロセスとすべての動詞を呼び出すことができます。

## 付記

【付記】 タブを使用して、自分または他の開発者が必要とする可能性がある情報を追加します。入力した付記は、設計時にのみ表示されます。ガイドまたはプロセスのユーザーやコンシューマに付記は表示されません。例えば、プロセスやガイドをデプロイする前に、実行する必要がある事柄についての通知を追加できます。

## プロセスのフィールド値の更新

更新リソースを使用して、既存のプロセスの次のフィールドの値を更新できます。

- 許可されたユーザー/許可されたグループ/匿名アクセスを許可
- API ゲートウェイからの HTTP 認証要求
- 実行日時
- トレースレベル

単一プロセスまたは複数のプロセスのフィールドを更新できます。更新リソースを使用してプロセスを更新するには、管理者、デプロイヤ、デザイナー、またはオペレータのロールが割り当てられている必要があります。

Postman などの REST クライアントで入力データとして更新するプロセスとフィールドを追加できます。入力データは、プロセスのグローバル識別子/グローバル意識別子 (GUID) と、更新されるデプロイメントのキーと値のペアで構成されます。GUID の詳細については、データ統合ヘルプの「*REST API リファレンス*」にある「*ルックアップリソース*」と「*アセット検索リソース*」を参照してください。

フィールド値を取得するには、次の URL を使用します。

```
<Informatica Intelligent Cloud Services URL>/active-bpel/asset/v1/update?assetType=Process
```

次のサンプルに示すように、入力データを追加できます。

```
PUT <baseApiUrl>/active-bpel/asset/v1/update?assetType=Process
```

```
Content-Type: application/json
```

```
Accept: application/json
```

```
INFA-SESSION-ID: <Infa-Session-ID>
```

```
{
  "processes": [
    {
      "guid": {
        "$t": "<guidvalue>"
      },
      "deployment": [
        {
          "key": {
            "$t": "Allow Anonymous Access"
          },
          "value": {
            "$t": "true"
          }
        },
        {
          "key": {
            "$t": "Run On"
          },
          "value": {
            "$t": "cloud server"
          }
        },
        {
          "key": {
            "$t": "Tracing Level"
          },
          "value": {
            "$t": "normal"
          }
        }
      ]
    }
  ]
}
```

```
}  
]  
}
```

ログインリソースを使用して、INFA-SESSION-ID を取得できます。ログインリソースの詳細については、データ統合ヘルプの「REST API リファレンス」を参照してください。

フィールドが正常に更新されると、次の応答を受け取ります。

```
{  
  "status": {  
    "$t": "SUCCESS"  
  }  
}
```

無効な Secure Agent を入力すると、次の応答が返されます。

```
{  
  "status": {  
    "$t": "FAILED"  
  },  
  "errorDetails": {  
    "error": {  
      "$t": "Update failed for assets"  
    },  
    "failedAssets": {  
      "updateAssetFaultInfo": {  
        "xmlns": "https://com.informatica.icrt/2021/08/updateAssets.xsd",  
        "xmlns$aetgt": "https://com.informatica.icrt/2021/08/updateAssets.xsd",  
        "name": {},  
        "GUID": {  
          "$t": "<Incorrect guid value>"  
        },  
        "fault": {  
          "$t": "The Secure Agent is not valid. Specify a valid Secure Agent."  
        }  
      }  
    }  
  }  
}
```

## プロセス内のフィールドを更新する際のルールとガイドライン

更新リソースを使用してプロセス内のフィールドを更新する場合は、次のルールとガイドラインを考慮してください。

- 「許可されたユーザー/許可されたグループ」フィールドまたは「匿名アクセスを許可」フィールドのいずれかを一度に設定する必要があります。そうしないと、エラーが発生します。
- API ゲートウェイからの HTTP 認証要求、または [匿名アクセスを許可] フィールドのいずれかを一度に設定する必要があります。そうしないと、エラーが発生します。
- API ゲートウェイからの HTTP 認証要求と [匿名アクセスを許可] フィールドでは、値を true または false として指定する必要があります。それ以外の値 (yes または no など) を指定すると、エラーが発生します。
- [トレースレベル] フィールドには、[なし]、[簡易]、[標準]、[詳細] のいずれかの値を指定する必要があります。そうしないと、エラーが発生します。
- API ゲートウェイからの HTTP 認証要求の値を true に設定する場合は、[次の設定で実行 (ランタイム環境)] フィールドの値を [クラウドサーバー] として指定する必要があります。[次の設定で実行] の値を [Secure Agent] として指定すると、エラーが発生します。
- パブリッシュ済みプロセスのランタイム環境の値を更新する場合は、まずプロセスをパブリッシュ解除する必要があります。



# プロセスステップの追加

ステップを追加して、データやサービスにアクセスしたり、関連するオーケストレーションアクティビティを実行したりします。

ステップを追加するときに、そのステップのプロパティも定義します。

次のいずれかのタスクを実行して、プロセスにステップを追加します。

- 左側のパレットからキャンバスにステップをドラッグします。
- キャンバスをダブルクリックし、[ステップのタイプ:] リストからステップを選択します。

## 割り当てステップ

フィールドに値を設定するには、割り当てステップを使用します。[割り当て] ステップを作成するには、プロセスをクリックし、[デザイン] キャンバスから [割り当て] を選択します。次に、[割り当て] のプロパティパネルで [追加] アイコンをクリックして、[名前] と [割り当て] の値を割り当てます。

次の図は、[割り当て] ステップを示します。

Field	Assigned Using	From
input1	Formula	math:sqrt(100)
temp1	Field	input1
input2	Content	100
temp2	Specific date	06/15/2021 12:00 PM

フィールドを追加するには、[追加] アイコンをクリックし、各フィールドに次の情報を追加します。

### 名前

これは、完全修飾のフィールド名です。名前には、英数字、アンダースコア(\_)、スペース、および Unicode 文字のみを含めることができます。

### 割り当て

これは、フィールドの値を取得するソースです。表示されるフィールドは、[開始] > [プロパティ] > [入力フィールド] または [一時フィールド] で定義したデータ型によって異なります。

例えば、入力フィールドのデータ型を [日付] または [時間] として定義すると、割り当てステップのそのフィールドに次のオプションが表示されます。

- 特定の日付
- 本日からの日数
- 前/後の日数
- フィールド
- 計算式

データ型を整数またはテキストとして定義した場合、フィールドの値を指定するための次のオプションが表示されます。

- コンテンツ
- フィールド
- 計算式

データ型の詳細については、「[\[フィールドのプロパティ\] ダイアログの使用](#)」(ページ 20)を参照してください。

## サービスステップ

サービスステップを追加するときに、いくつかのプロパティを設定します。

次のセクションでは、サービスステップのプロパティについて説明しています。

### 全般

プロパティ	説明
ステップのタイプ	サービスステップ。
名前	サービスを説明する名前。名前には、英数字、アンダースコア(_)、スペース、および Unicode 文字のみを含めることができます。

### サービス

プロパティ	説明
サービスタイプ	プロセスに追加する接続、プロセス、またはシステムサービス。既存のタスクの一覧から選択します。 注: プロセスに追加する既存の項目が必要です。プロセスを作成するときに、タスクは作成できません。

システムサービスについてサポートされているアクションのリストの詳細については、「[システムサービスアクション](#)」(ページ 86)を参照してください。

サービスステップにサービスを追加するときに、対応する入力フィールドが作成されます。

### 入力フィールド

[入力フィールド] セクションに、この種類のステップを使用するときに最も多く使用される入力フィールドの名前が表示されています。オブジェクトを作成するサービスステップの場合、表示される入力フィールドは、オブジェクトの作成時に最も頻繁に必要なフィールドになります。(必要でないオプションのフィールドは、削除できます。) 追加の (オプション) 入力フィールドをリストから選択できます。

削除アイコンを使用して、フィールドを削除します。これにより、サービスステップに渡さない入力フィールドが削除されます。一部のフィールドは必須で、削除できません。

### フォールト処理

Process Designer のフォールト処理は、ビジネスプロセスモデリング表記法 (BPMN) 2.0 の仕様に基づいています。これは、境界イベントの概念を定義します。境界イベントは、全体的なプロセスの範囲ではなく、特定のステップに関連付けられたフォールトを取得します。これは、フォールトの処理を、プロセスレベルではなく、ステップレベルで行えることを意味します。(開発者は、プロセススコープのフォールトハンドラを定義するためにも Process Developer を使用できます。)

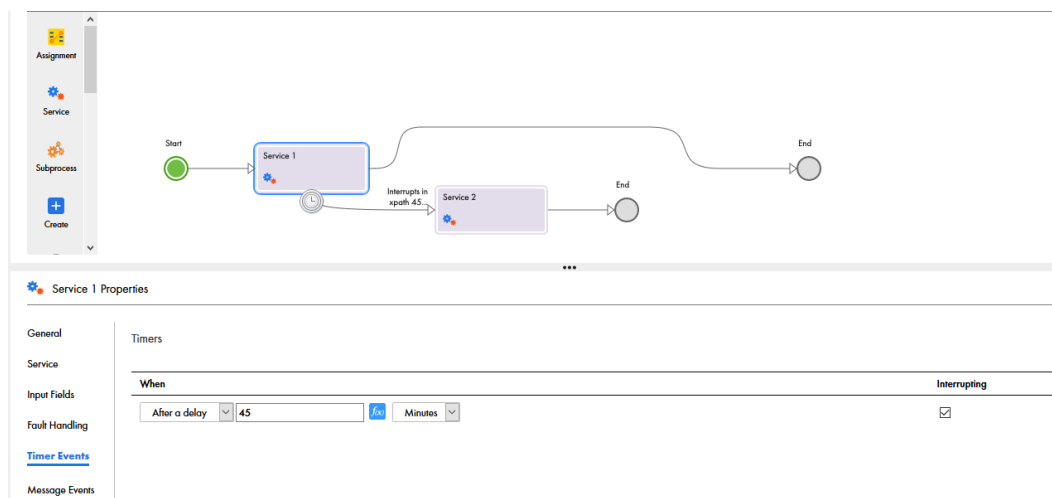
## アプリケーション統合コンソールでのフォルトの可視性

プロセスでフォルト処理が有効になっている場合、アプリケーション統合コンソールのプロセスリストに、ステップ上のエラーマーカーと、いくつかの基本的なフォルト情報が表示されます。

## タイマイベント

タイマイベントを使用して、スケジュールに基づいてアクションを実行します。イベントを【指定時刻】に実行するか、【遅延後】に実行するかを指定できます。

次の図は、設定したタイマイベントを示します。



タイマでサービスステップを中断する場合は、【中断】を選択します。中断タイマを設定すると、サービスステップタスクは中断されて、プロセスは設定されたタイマでのみタスクを実行します。

## メッセージイベント

メッセージプロバイダと関連で使用するメッセージを指定します。

## サブプロセスステップ

サブプロセスステップは、あるプロセスを別のプロセスの中に組み込みます。サブプロセスステップが実行されると、組み込みプロセスが実行されます。

多数のステップを含むプロセスがある場合は、オーケストレーションロジックを複数の小さなプロセスに分割することを検討してください。次に、サブプロセスステップを使用して、より小さなプロセスを親プロセスに組み込むことにより、設計を簡略化できます。これは、モジュラー型設計につながるだけでなく、編集用にプロセスを開いたときの読み込みを高速化するのにも役立ちます。

サブプロセスステップ名には、英数字、アンダースコア ( \_ )、スペース、および Unicode 文字のみを含めることができます。

サブプロセスステップに対し、該当する組み込みプロセスがある場合、それに基づいて入力および出力関連フィールドを定義できます。

次の図に、JMS キュー削除サブプロセスのプロパティを示します。

JMSDequeue Properties

General

Subprocess

Iteration Rule

Input Fields

Fault Handling

Timer Events

Message Events

Process: JMSDequeue Select...

Description  
No description

Input Fields

Name	Required	Type	Description
in	<input type="checkbox"/>	Object ID (Process Objects:TestPO)	

Output Fields

Name	Type	Description
out	Object ID (Process Objects:TestPO)	

## 反復ルール

反復ルールは、サブプロセスステップで選択したプロセスの【適用対象】フィールドがオブジェクトタイプに設定されている場合に適用されます。

単一のオブジェクトまたは指定したリスト内のすべてのオブジェクトで実行するようにサブプロセスを設定できます。

【**単一オブジェクトで実行**】を選択した場合、サブプロセスは、入力として受け取る特定のオブジェクト上で実行されます。実行が完了すると、サブプロセスはサブプロセスの次のステップに制御を渡します。

【**次のタイミングまでリスト内の各オブジェクトで実行**】を選択した場合は、サブプロセスの実行を制御するイベントも選択します。

次のいずれかの値を選択して、実行時にサブプロセスの実行を制御するイベントを定義できます。

- **【フィールド】**：両方のオプションで利用可能で、【**単一のオブジェクトで実行**】と【**リスト内の各オブジェクトで次まで実行**】、つまり、選択した実行の選択に基づいて、単一のオブジェクトまたはオブジェクトのリストがサブプロセスに渡される場合です。これには、接続プロパティおよび入力パラメータで使用可能なプロセスオブジェクトまたは Salesforce 接続フィールドが含まれます。
- **【クエリ】**：【**単一のオブジェクトで実行**】オプションを選択した場合にのみ使用可能。サブプロセスに渡されるオブジェクトを選択するための WHERE 条件を定義できます。
- **【リスト】**：【**リスト内の各オブジェクトで次まで実行**】オプションを選択した場合にのみ使用可能。サブプロセスに渡されるオブジェクトを選択するための WHERE 条件を定義できます。

次の図は、反復ルールのプロパティを示しています。

JMSDequeue Properties

General

Subprocess

**Iteration Rule**

Input Fields

Fault Handling

Timer Events

Message Events

Applies To: JMSDequeue

Run on a single object

Run on each object in list until End of list

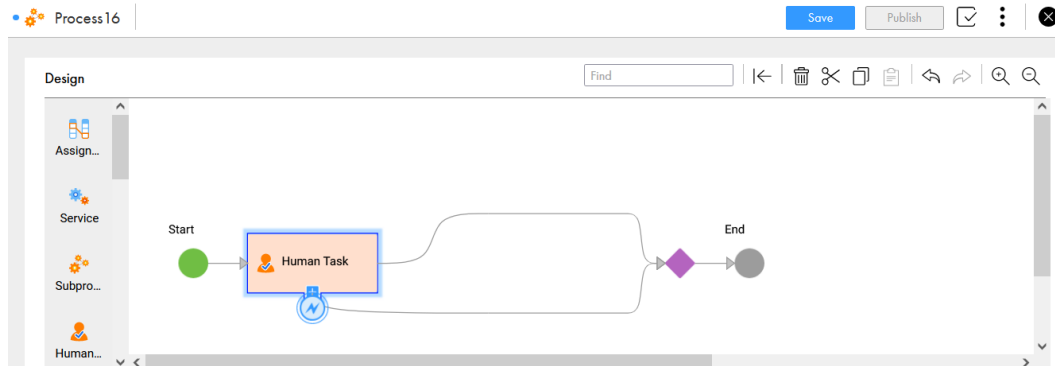
On Object List: Field untitled1

## ヒューマンタスクステップ

ヒューマンタスクステップを使用して、アプリケーション統合プロセスに人間による意思決定を組み込みます。承認や例外の管理には人間のアクションが必要です。例えば、ローン承認プロセスにヒューマンタスクステップを追加するとします。

ヒューマンタスクをプロセスに組み込むには、ヒューマンタスクステップをパレットからキャンバスにドラッグアンドドロップします。ヒューマンタスクステップをダブルクリックし、ヒューマンタスクアセット内に作成されたヒューマンタスクを選択します。

次の図は、アプリケーション統合プロセス内のヒューマンタスクステップを示しています。



次の表に、ヒューマンタスクステップで設定できるプロパティを示します。

プロパティ	説明
全般	ヒューマンタスクステップの名前。
ヒューマンタスク	アプリケーション統合プロセスに組み込むヒューマンタスクを選択します。
入力フィールド	ヒューマンタスクの入力フィールドに値を入力します。
結果	ヒューマンタスクアセットで定義した結果を含むリストが表示されます。
フォールト処理	[キャッチフィールド] オプションを選択して、ヒューマンタスクステップのフォールト処理を有効にします。[フォールトフィールド名] フィールドでフォールトハンドラの名前を指定できます。

ヒューマンタスクアセットの作成の詳細については、「[ヒューマンタスクの作成](#)」(ページ 280)を参照してください。

## 作成ステップ

作成ステップを使用して、新しいオブジェクトインスタンスを追加します。例えば、アカウントオブジェクトを操作する場合、作成ステップを使用して、新しいアカウントを追加できます。

作成ステップ名には、英数字、アンダースコア ( \_ )、スペース、および Unicode 文字のみを含めることができます。

次の図に、アカウントオブジェクトの作成ステップを示します。

Design Find

---

**+** Account Properties

---

General

Connection:

Object:

Description

Create a step type Account.

Input Fields

Fault Handling

Name	Required	Type
Account Name	<input checked="" type="checkbox"/>	Text
Account Type	<input type="checkbox"/>	Picklist
Parent Account ID	<input type="checkbox"/>	Object ID (Salesforce:Account)
Billing Street	<input type="checkbox"/>	Text Area
Billing City	<input type="checkbox"/>	Text
Billing State/Province	<input type="checkbox"/>	Text
Billing Zip/Postal Code	<input type="checkbox"/>	Text
Billing Country	<input type="checkbox"/>	Text

新しいオブジェクトを作成する手順

1. **【入力フィールド】** タブを選択し、オブジェクトに追加するすべてのフィールドを選択します。必須フィールドとオプションフィールドがあります。
2. 各フィールド名に対し、ソースを **【コンテンツ】**、**【フィールド】**、**【計算式】** のいずれかの値に設定します。

### 参照フィールドの使用

以下の点に注意してください。

- 作成するオブジェクトに対する参照フィールドを **【入力フィールド】** タブに含めない場合、Process Designer は、現在のオブジェクトの値を実行時に使用します。
- **【入力】** タブで参照フィールドが明示的に設定されない場合、**【適用対象】** オブジェクトの参照フィールドが設定されます。
- 参照フィールドのコンテンツが空に設定された場合、参照フィールドは作成文には含められません（これにより、明示的にフィールドが空に設定された場合、または NULL 値がデータの制約に違反する可能性がある場合に問題が発生することが防止されます）。

## 受信ステップ

受信ステップは、サービスを行うプロセス内で定義でき、プロセスを続行する前に、なんらかのイベントデータを待機します。例えば、注文プロセスに、注文が特定のしきい値を超えたときにクレジット情報をさらに要求するステップが含まれている場合、受信ステップを使用することで、受信呼び出しのブロックを処理できます。

The screenshot shows a BPMN editor interface. On the left is a palette with icons for Subprocess, Create, Receive, Wait, Milestone, and End. The main canvas displays a process flow: Start (green circle) → Receive 1 (blue circle with envelope icon) → End (grey circle). Below the canvas, the 'Receive 1 Properties' dialog is open. The 'Receive' tab is selected. Under 'General', the 'Message' dropdown is set to 'MessageEvent 1'. Under 'Correlation', the 'InFieldMsgEvent' dropdown is set to 'Literal', and the text input field contains '<Give values>'.

受信ステップ名には、英数字、アンダースコア ( \_ )、スペース、および Unicode 文字のみを含めることができます。

受信ステップの出力は、サービス内の出力フィールドと同様に、プロセスの後半で使用できます。

**相関割り当て**では、着信メッセージイベントと相関させるフィールドを選択します。これらの割り当てを使用して、要求の実行をキャンセルする（中断）か、ステータスを提供する（非中断）ことができます。

詳細については、「[相関メッセージイベント](#)」（[ページ 67](#)）を参照してください。

## 待機ステップ

待機ステップを追加するときに、いくつかのプロパティを設定します。

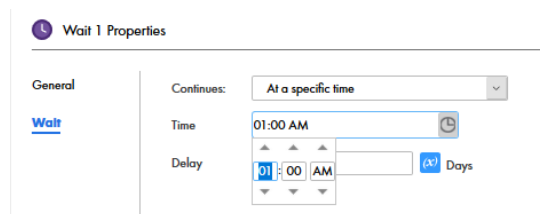
次の表では、待機ステップのプロパティについて説明しています。

プロパティ	説明
名前	待機ステップの名前。名前には、英数字、アンダースコア ( _ )、スペース、および Unicode 文字のみを含めることができます。
待機	プロセスが一時停止する時刻と長さを決定するプロパティ。次のオプションから選択できます。 <ul style="list-style-type: none"> <li>- 特定の時間</li> <li>- 待機期間後</li> </ul>

## 特定時点での一時停止

このオプションを選択して、プロセスを特定の時間に一時停止します。プロセスを一時停止する【時間】を入力し、必要に応じて【遅延】を指定します。【遅延】の値には、整数または定義したフィールドを指定できません。

次の画像は、【特定の時間】オプションを示しています。



Wait 1 Properties

General

Wait

Continues: At a specific time

Time: 01:00 AM

Delay: 01:00 AM Days

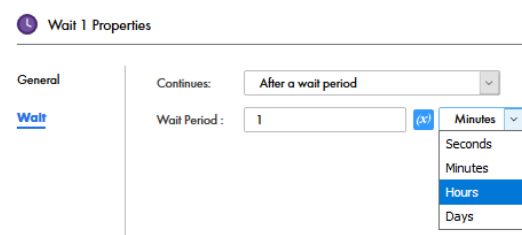
例えば、3日後の午前1時に一時停止するようにプロセスを設定するとします。午前1時は【時間】であり、3日は【遅延】です。

## 待機期間後

このオプションを選択して、一定期間の経過後にプロセスを一時停止します。この期間は、プロセスが待機ステップに到達した時点で開始されます。プロセスを一時停止する【待機期間】を入力します。【待機期間】の値には、定義した整数またはフィールドを指定できます。

例えば、プロセスが待機ステップに到達した時点から1時間、プロセスを一時停止するように設定したりできます。

次の画像は、【待機期間後】オプションを示しています。



Wait 1 Properties

General

Wait

Continues: After a wait period

Wait Period: 1 Minutes

Seconds

Minutes

Hours

Days

## マイルストーンステップ

マイルストーンステップを使用して、プロセスの終了時に2つのアクションのいずれかを指定できます。

次の図に、[マイルストーン] ステップを示します。



Milestone 1 Properties

General

Milestone

Ending Type:  Milestone: Send a Synchronous Response  End of Process

HTTP Status: 200 OK

HTTP Response Headers (1)

Name	Content
Identifier	\$input.CurrentUser.FederationIdentifier

次のマイルストーンステップのプロパティを設定できます。

### 名前

マイルストーンステップの名前が表示されます。この値は編集できます。



## 終了タイプ

次の値から選択することができます。

### マイルストーン: 同期応答の送信

同期応答を元の要求に送信します。この応答には、[プロセスのプロパティ] ダイアログで定義されたすべての出力フィールドが含まれます。同期応答がすでに送信されている場合、アクションは発生しません。これはエラーではなく、バックグラウンド処理とバッチ処理が発生するようにするための仕様です。マイルストーンステップの後に追加のステップを実行して同期メッセージを送信し、実行を続行できます。

**注:** マイルストーンステップを並列パスステップに挿入する場合は、[終了タイプ] が [マイルストーン: 同期応答の送信] に設定されている必要があります。

### プロセスの終了

プロセスが終了すると、その後のアクションは発生しません。

## HTTP ステータス

HTTP 応答のステータスコードが表示されます。デフォルト値は 200 OK です。この値は編集できます。

## HTTP 応答ヘッダー

プロセスに関連付けられた REST API がプロセスの実行後に返す必要がある HTTP 応答ヘッダーと値を指定します。

例えば、組織 ID を返すように応答ヘッダーを設定できます。プロセスがダウンストリームアプリケーションで返すヘッダー値を使用できます。

ルックアップ値を返すように応答ヘッダーを設定できます。プロセスが返すルックアップ値に基づいてディビジョンを作成し、ダウンストリームアプリケーションでさらに処理を実行することができます。

X-XSS-Protection HTTP 応答ヘッダーを設定して、さまざまなブラウザで REST API の追加のセキュリティを有効にすることができます。

ヘッダー名として X-XSS-Protection と入力し、値として 1; mode=block を入力します。

X-XSS-Protection ヘッダーにより、クロスサイトスクリプティング (XSS) のフィルタリングが有効になります。ブラウザが XSS 攻撃を検出した場合、ページはレンダリングされません。

各ヘッダーに名前と値を設定することで、プロセスに複数の応答ヘッダーを追加できます。ヘッダー名は一意である必要があり、次の文字は使用できません。

( ) < > @ , ; \\ / [ ] ? = { } \_

また、ヘッダー名には英字以外の文字を含めないでください。ヘッダー値に英語以外の文字を指定する場合は、テキストを Base64 形式でエンコードする必要があります。ダウンストリームアプリケーションは、元のコンテンツを取得するためにテキストをデコードする必要があります。手動で値を追加するか、計算式を入力できます。

プロセスは、次のデフォルトの HTTP 応答ヘッダーを使用します。

- access-control-allow-credentials
- cache-control
- content-encoding
- content-type
- date
- expires
- keep-alive

- ステータス
- strict-transport-security
- transfer-encoding
- vary
- x-frame-options
- x-http-status
- x-instanceid

デフォルトのヘッダー名と値を上書きすることはできません。

プロセスの実行後、API は設定された HTTP 応答ヘッダーを返します。応答ヘッダーは、アプリケーション統合の **【自分のプロセス】** ページ、またはアプリケーション統合コンソールの **【プロセス】** ページで確認できます。

## 終了ステップ

終了ステップは、プロセスの終了を示します。実行がこのステップに達すると、プロセスは完了します。

次の終了ステップのプロパティを設定できます。

### 名前

終了ステップの名前が表示されます。この値は編集できます。

### 終了タイプ

デフォルト値は End of Process です。この値は編集できません。

### HTTP ステータス

HTTP 応答のステータスコードが表示されます。デフォルト値は 200 OK です。この値は編集できます。

### HTTP 応答ヘッダー

プロセスに関連付けられた REST API がプロセスの実行後に返す必要がある HTTP 応答ヘッダーと値を指定します。

例えば、組織 ID を返すように応答ヘッダーを設定できます。プロセスがダウンストリームアプリケーションで返すヘッダー値を使用できます。

ルックアップ値を返すように応答ヘッダーを設定できます。プロセスが返すルックアップ値に基づいてディビジョンを作成し、ダウンストリームアプリケーションでさらに処理を実行することができます。

X-XSS-Protection HTTP 応答ヘッダーを設定して、さまざまなブラウザで REST API の追加のセキュリティを有効にすることができます。

ヘッダー名として X-XSS-Protection と入力し、値として 1; mode=block を入力します。

X-XSS-Protection ヘッダーにより、クロスサイトスクリプティング (XSS) のフィルタリングが有効になります。ブラウザが XSS 攻撃を検出した場合、ページはレンダリングされません。

各ヘッダーに名前と値を設定することで、プロセスに複数の応答ヘッダーを追加できます。ヘッダー名は一意である必要があり、次の文字は使用できません。

( ) < > @ , ; \\ / [ ] ? = { } \_

また、ヘッダー名には英字以外の文字を含めないでください。ヘッダー値に英語以外の文字を指定する場合は、テキストを Base64 形式でエンコードする必要があります。ダウンストリームアプリケーションは、元のコンテンツを取得するためにテキストをデコードする必要があります。手動で値を追加するか、計算式を入力できます。

プロセスは、次のデフォルトの HTTP 応答ヘッダーを使用します。

- access-control-allow-credentials
- cache-control
- content-encoding
- content-type
- date
- expires
- keep-alive
- ステータス
- strict-transport-security
- transfer-encoding
- vary
- x-frame-options
- x-http-status
- x-instanceid

デフォルトのヘッダー名と値を上書きすることはできません。

プロセスの実行後、API は設定された HTTP 応答ヘッダーを返します。応答ヘッダーは、アプリケーション統合の【自分のプロセス】ページ、またはアプリケーション統合コンソールの【プロセス】ページで確認できます。

## スローステップ

スローステップを使用して、エラーペイロードを構成できます。

次の表に、スローステップ内のプロパティを示します。

プロパティ	説明
名前	必須。スローステップを説明する名前を入力します。名前には、英数字、アンダースコア(_)、スペース、および Unicode 文字のみを含めることができます。
コード	必須。faultInfo のペイロードの名前を格納する文字列値です。
明細	オプション。faultInfo の詳細を格納する文字列値です。
理由	オプション。faultInfo の理由詳細です。これは\$any 型にできます。

## 決定ステップ

決定ステップを使用すると、プロセスが、フィールドの値に応じて異なるパスを取ることが可能になります。以下の表に、決定ステップのプロパティを示します。

プロパティ	説明
名前	決定ステップの名前。名前には、英数字、アンダースコア(_)、スペース、および Unicode 文字のみを含めることができます。
決定	<p>プロセスは、ここで定義したフィールドとパスに基づいて決定を行います。</p> <p><b>【スタート】</b> &gt; <b>【フィールド】</b> で定義したフィールドの一覧からフィールド名を選択します。決定ステップで決定を下すために必要な条件と値を入力します。</p> <p>使用できる条件は、選択したフィールドによって異なります。</p> <p>例えば、<b>【簡易】</b> &gt; <b>【テキスト】</b> を選択した場合、次の条件が使用できます。</p> <ul style="list-style-type: none"><li>- 次の値に等しい</li><li>- 次で開始する</li><li>- 次で終わる</li><li>- 次のいずれかで開始する</li><li>- 次を含む</li></ul> <p>選択した条件に対してテキスト値を入力できます。</p> <p>決定ステップには複数の条件を追加できます。各条件は、潜在的なデータパスです。</p> <p>追加したパスごとに、対応する分岐が UI に表示されます。分岐をドラッグして、分岐が UI に表示される順序を並べ替えます。</p> <p>ほとんどの決定ステップには、別のパスがあります。テストの条件を満たしているデータがない場合、このパスが実行を処理します。</p>

### パスの評価

プロセスは、指定した条件に基づいて条件を評価します。パスは必ず、交差しない条件を指定して作成します。

例えば、次のパスでデータ決定ステップを作成します。

- パス 1: フィールドは 100 以下とする。
- パス 2: フィールドは 75 以下とする。
- パス 3: フィールドは 25 以下とする。
- パス 4: それ以外。

データ決定ステップが作成された整数フィールドの値が 25 の場合、データ決定ステップはパス 1 を受け取ります。これは 25 が 100 未満で、パス 1 が最初のオプションであるためです。

データ決定ステップが「25 未満のフィールド以下」のパスに従っていることを確認するには、次の条件でパスを再作成します。

- パス 1: 0 から 25 の間の整数。
- パス 2: 26 から 75 の間の整数。
- パス 3: 76 から 100 の間の整数。
- パス 4: それ以外。

**重要:** プロセスは、トップダウン方式で条件を評価します。それ以外の分岐が最後のパスであることを確認します。

決定ステップは、別の決定ステップに連結できます。例えば、年間所得が 10 万ドルを超えると、分岐を実行できるなどと指定できます。同じパスの次の決定テストで、都市がボストンであるか、それ以外であるかをテ

ストできます。この方法を使用するときには、最初の条件での真の分岐に基づいて 2 番目の条件をテストするため、ブール値 AND ロジックを使用します。この例では、決定ステップを使用して、「年間収入が 10 万ドルを超え、都市はボストンである」という AND 条件を設定します。

同様に、ブール型 OR ロジックをサポートするために、任意の分岐に 2 番目の条件のテストを追加できます。

## 並列パスステップ

並列パスステップを追加するとき、いくつかのプロパティを設定します。

以下の表に、並列パスステップのプロパティを示します。

プロパティ	説明
名前	並列パスステップの名前。名前には、英数字、アンダースコア(_)、スペース、および Unicode 文字のみを含めることができます。
並列パス	<p>プロセスで並列して実行するパス。</p> <p><b>注:</b> 並列パスステップを使用した場合でも、すべてのブランチのすべてのステップが同時に開始されるわけではありません。すべてのパスのステップの開始時間にはタイムラグがあります。ブランチの実行は、各ブランチで使用されるステップによって異なります。</p> <p>複数のブランチが順番に実行されている場合、1つのブランチが特定の時間一時停止すると、プロセスはステップが完了するまでそのブランチで待機するのではなく、別のブランチからのステップの実行を並行して開始します。例えば、ブランチに待機ステップがあり、プロセスが特定の時間一時停止した場合は、別のブランチのステップの実行が並行して開始されます。</p> <p><b>[追加]</b> をクリックして新しい分岐を追加します。</p> <p>各分岐に複数のステップを追加できます。分岐にステップを追加するには、左側のパレットからステップをドラッグアンドドロップします。</p> <p>ジャンプステップを並列パスステップと連動させて使用した場合は、同じ並列パスブランチ内の別のステップにのみジャンプできます。</p> <p>ジャンプステップと並列パスステップを同時に使用する場合は次の制限を念頭に置いておいてください:</p> <ul style="list-style-type: none"> <li>- 並列パスステップ上にいる場合は、同じ並列パスステップの別のブランチのステップにジャンプすることはできません。</li> <li>- 並列パスステップ上にいる場合は、並列パスステップ外のいかなるステップへもジャンプできません。</li> <li>- 並列パスステップの外にいる場合は、並列パスステップ内のいかなるステップへもジャンプすることはできません。</li> </ul>

## ジャンプステップ

ジャンプステップを追加するときに、いくつかのプロパティを設定します。

以下の表に、ジャンプステップのプロパティを示します。

プロパティ	説明
送信先	<p>ジャンプのターゲット。使用可能なステップの一覧から選択します。</p> <p>複数のステップが同じターゲットステップにジャンプできます。ターゲットとして特定のステップがあるジャンプステップの数を確認するには、ターゲットステップの横にある矢印の上にカーソルを置きます。</p> <p>ジャンプステップを並列パスステップと連動させて使用した場合は、同じ並列パスブランチ内の別のステップにのみジャンプできます。</p> <p>ジャンプステップと並列パスステップを同時に使用する場合は次の制限を念頭に置いてください:</p> <ul style="list-style-type: none"><li>- 並列パスステップ上にいる場合は、同じ並列パスステップの別のブランチのステップにジャンプすることはできません。</li><li>- 並列パスステップ上にいる場合は、並列パスステップ外のいかなるステップへもジャンプすることはできません。</li><li>- 並列パスステップの外にいる場合は、並列パスステップ内のいかなるステップへもジャンプすることはできません。</li><li>- フォールトパスからはいかなるステップへもジャンプすることはできません。ジャンプステップを追加するには、フォールトパスをメインプロセスパスにマージする必要があります。</li><li>- フォールトパスの外にいる場合は、フォールトパス内のステップにジャンプすることはできません。</li></ul> <p>プロセス内でジャンプステップを使用する場合は、次のベストプラクティスを使用してください。</p> <ul style="list-style-type: none"><li>- 外部フローからフォールトパス内のステップへのジャンプステップを使用する代わりに、ジャンプステップをフォールトパス内と同じフローに置き換えてプロセスを設計します。</li><li>- 反復する目的でジャンプステップを使用し、そのステップがフォールト処理を含むサービスステップまたはサブプロセスステップを指している場合は、フォールト処理を含むサービスステップまたはサブプロセスステップを別のサブプロセスステップ内に囲みます。</li></ul>

## システムサービスアクション

プロセスでサービスステップを設定し、サービスタイプとしてシステムサービスを選択するときに、サポートされるアクションのリストから選択できます。

### オブジェクトの削除

オブジェクトの削除アクションを使用すると、レコードを削除できます。アプリケーションに応じて、削除する対象は、オブジェクトまたはオブジェクトインスタンスと呼ばれます。

**Delete Object Properties**

Service Type: System Service

Action: Delete Object

Description: Deletes an object.

Input Fields

Name	Required	Type	Description
Connection Name	<input checked="" type="checkbox"/>	Text	Name of the connection to which the object belongs to.
Object Type	<input checked="" type="checkbox"/>	Text	Object type of the object to delete.
Object Id	<input checked="" type="checkbox"/>	Object ID	Id of the object to delete.

Output Fields

None

図に示すように、3つのフィールドにこれらの情報を入力する必要があります。

## Data Marketplace API の実行

Data Marketplace API を使用すると、GET、POST、PUT などのメソッドを使用してデータアセットやデータコレクションにアクセスし、ビジネスオーケストレーションを実行できます。Data Marketplace API は、クラウドサーバーで実行されるプロセスに対して実行できます。

Data Marketplace API は、サービスステップで設定し、API ゲートウェイを使用して呼び出すことができます。

プロセスで Data Marketplace API を使用するには、**[許可されたグループ]** フィールドまたは **[許可されたユーザー]** フィールドを設定する必要があります。Data Marketplace API と統合するプロセスを呼び出すには、Data Marketplace の管理者ロールが必要です。

次の図は、[Data Marketplace API の実行] のプロパティを示しています。

**Execute Data Marketplace API Properties**

Service Type: System Service

Action: Execute Data Marketplace API

Description: Execute the specified Data Marketplace API.

Input Fields

Name	Required	Type	Description
HTTP Method	<input checked="" type="checkbox"/>	Text	The HTTP method to invoke the Data Marketplace API.
Relative Path	<input checked="" type="checkbox"/>	Text	The relative path of the Data Marketplace API.
Payload	<input type="checkbox"/>	Text	The JSON payload to be sent to the Data Marketplace API.
Query Parameter	<input type="checkbox"/>	Object ID	The process object type field associated with the Data Marketplace API query parameters.
Path Variable	<input type="checkbox"/>	Object ID	The process object type field associated with the Data Marketplace API path variables.

Output Fields

Name	Type
Data Marketplace Response	Text

### 入力フィールド

Data Marketplace API のサービスステップで、次の入力フィールドを設定します。

#### HTTP メソッド

必須。GET、PUT、POST などの Data Marketplace API を呼び出す HTTP メソッド。

## 相対パス

必須。Data Marketplace API の相対パス。

例えば、Data Marketplace のデータアセットに対して操作を実行するには、相対パスとして次を入力します。

```
dataAssets
```

Data Marketplace のデータ要素に対して操作を実行するには、相対パスとして次を入力します。

```
dataAssets/{id}/dataElements
```

## ペイロード

PUT または POST メソッドを使用して Data Marketplace API に送信される JSON ペイロード。

例えば、従業員データを Data Marketplace API に送信するには、次のサンプルに示すように JSON 形式でデータを入力します。

```
{
  "Employee": [
    {
      "firstName": "Cindy",
      "lastName": "Louis",
      "department": "IT",
      "status": "Employed"
    }
  ]
}
```

## クエリパラメータ

Data Marketplace API のクエリパラメータに関連付けられたデータを返すプロセスオブジェクト。

sort や status などの API クエリパラメータを使用し、GET メソッドを使用してデータを並べ替えたりフィルタリングしたりできます。API は、クエリパラメータに基づいて、関連付けられたデータを XML 形式で返します。

[クエリパラメータ] フィールドは、計算式またはフィールドを使用して設定できます。

例えば、計算式を使用して、アセットの最大数と有効なステータスを取得するには、次のサンプルに示すように status クエリパラメータを使用します。

```
<DataMarketplaceQueryParameters>
  <queryParams>
    <name>status</name>
    <value>ENABLED</value>
  </queryParams>
  <queryParams>
    <name>limit</name>
    <value>10</value>
  </queryParams>
</DataMarketplaceQueryParameters>
```

他のフィールドを使用して [クエリパラメータ] フィールドを設定する方法については、[「Data Marketplace API を実行するためのフィールドの作成」](#) (ページ 89) を参照してください。

## パス変数

Data Marketplace API のパス変数に関連付けられたプロセスオブジェクト。

[パス変数] フィールドは、計算式またはフィールドを使用して設定できます。

例えば、データ要素をデータアセットに挿入するには、次のサンプルに示すように、パス変数に ID を指定します。

```
<DataMarketplacePathVariables>
  <pathVariables>
```



```

    <name>id</name>
    <value>eu33-39839j2</value>
  </pathVariables>
</pathVariables>
</DataMarketplacePathVariables>

```

他のフィールドを使用して [パス変数] フィールドを設定する方法については、[「Data Marketplace API を実行するためのフィールドの作成」](#) (ページ 89) を参照してください。

## 出力フィールド

Data Marketplace API の割り当てステップまたは決定ステップで、次の出力フィールドを設定します。

名前	タイプ	説明
Data Marketplace の応答	テキスト	出力を格納する文字列変数。

Data Marketplace API の詳細については、Data Marketplace のヘルプを参照してください。

## Data Marketplace API を実行するためのフィールドの作成

クエリパラメータとパス変数は、値をフィールドとして渡すことで設定できます。

クエリパラメータとパス変数の値をフィールドとして渡すには、一時フィールドを作成する必要があります。次の図に示すように、カスタムタイプとして、クエリパラメータフィールドには **[Data Marketplace のクエリパラメータ]** プロセスオブジェクト、パス変数フィールドには **[Data Marketplace のパス変数]** プロセスオブジェクトを選択します。

Name	Type	Description	Initial Value
Query Parameter 1	Data Marketplace Query Parameters		
Path Variable 1	Data Marketplace Path Variables		

作成した一時フィールドが割り当てステップに表示されます。割り当てステップでは、次の図に示すように、クエリパラメータとパス変数の一時フィールドの値を設定できます。

Field	Assigned Using	From	Status
Query Parameter 1 > queryParams > Name	Content	status	<input checked="" type="checkbox"/>
Query Parameter 1 > queryParams > Value	Content	Enabled	<input checked="" type="checkbox"/>
Path Variable 1 > pathVariables > Name	Content	ID	<input checked="" type="checkbox"/>
Path Variable 1 > pathVariables > Value	Content	eu33-39839j2	<input checked="" type="checkbox"/>

ここで、queryParams と pathVariables は、[Data Marketplace のクエリパラメータ] と [Data Marketplace のパス変数] プロセスオブジェクトで設定された、名前と値のペアを持つ入力フィールドです。

サービスステップでは、次の図に示すように、クエリパラメータまたはパス変数のフィールド値用に追加された一時フィールドを選択する必要があります。

Name	Required	Value	
HTTP Method	<input checked="" type="checkbox"/>	Content	GET
Relative Path	<input checked="" type="checkbox"/>	Content	dataAssets
Payload	<input type="checkbox"/>	Formula	{ ... }
Query Parameter	<input type="checkbox"/>	Field	Query Parameter 1
Path Variable	<input type="checkbox"/>	Field	Path Variable 1

## JMS キュー追加サービス

JMS キュー追加サービスを使用して、キューへの追加先 JMS サービスにメッセージを送信します。受信プロセスが準備完了すると、サービスはそのプロセスにメッセージを送信します。このメッセージは、Informatica Secure Agent を使用して送信され、オンプレミスサービスと通信します。

メッセージのデータには、次のコンポーネントが含まれています。

### Messaging Manager 名

宛先を含む Messaging Manager の名前。

### 宛先名

メッセージの宛先の名前。

### 形式

宛先にシリアル化される形式。

### メッセージ

宛先に送信されるメッセージ。

### ラップなしルート要素

プロセスオブジェクトが宛先に送信されるときシリアル化方法を記述します。

次の図に、JMS キュー追加サービスのプロパティを示します。

Name	Required	Type	Description
Messaging Manager Name	<input checked="" type="checkbox"/>	Text	Name of the Messaging Manager which contains the Destination.
Destination Name	<input checked="" type="checkbox"/>	Text	Name of the destination for the message.
Message	<input checked="" type="checkbox"/>	Text	The message to be send to the destination.
Format	<input checked="" type="checkbox"/>	Picklist	The format serialized to the destination.
Unwrap Root Element	<input type="checkbox"/>	Checkbox	Unwraps the contents of the root element. Applies to JSON and XML Formats only.

## JMS エンキューサービスのロギング

デフォルトでは、JMS エンキューサービスはエラーのみをログに記録します。メッセージ処理のフロー全体に対して追加のログを有効にするには、次の手順を実行します。

1. 次のディレクトリから log4j2.xml ファイルにアクセスします。  
<Secure Agent installation directory>\apps\process-engine\<latest\_version>\conf
2. 次のコードを追加します。  

```
<Logger name="org.apache.camel.component.file.remote" level="DEBUG" additivity="false">  
  <AppenderRef ref="catalina"/>  
</Logger>
```
3. log4j2.xml ファイルを保存します。

## MDM ヒューマンタスク

Informatica Intelligent Cloud Services Business 360 コンソールは、アプリケーションの統合プロセスを、ユーザーがトリガーしたビジネスイベントとシステムが生成したビジネスイベントのワークフローとして使用します。Business 360 コンソールで定義済みの承認ワークフローを使用しない場合は、アプリケーションの統合でプロセスを定義し、そのプロセスを使用してビジネスイベントを設定します。これらのプロセスには、少なくとも 1 つのヒューマンタスクが含まれている必要があります。

ヒューマンタスクは、プロセスがワークフローの一部としてユーザーインタラクションを必要とすることを示します。MDM ヒューマンタスクは、次のユーザータスクをサポートします。

- 汎用ユーザータスク。ワークフローに含まれるユーザーのアクションを定義します。
- オリジネータユーザータスク。アクションを使用してワークフローをトリガするユーザーのアクションを定義します。

定義済みの MDM マルチステップ承認テンプレートを使用して、プロセスを設計できます。定義済みのテンプレートの詳細については、[MDM Multi-Step Approval Template](#) を参照してください。

サービスステップで MDM ヒューマンタスクを設定するには、システムサービスアクションとして **[MDM] > [HumanTask] > [汎用ユーザータスク]** または **[MDM] > [HumanTask] > [オリジネータユーザータスク]** を選択します。

**注:** **[HumanTask]** にある **[汎用ユーザータスク]** と **[オリジネータユーザータスク]** 以外のオプションは、将来使用するために予約されています。

次の図は、MDM ヒューマンタスクのプロパティを示しています。

Generic User Task Properties

General

Service Type: System Service

Action: MDM > HumanTask > Generic User Task

Description  
No description

Input Fields

Name	Required	Type
Template	<input checked="" type="checkbox"/>	Object_ID_(Process_Objects:Template)
TaskTypes	<input checked="" type="checkbox"/>	Object_ID_(Process_Objects:TaskTypes)
Change List	<input checked="" type="checkbox"/>	Text
TaskName	<input checked="" type="checkbox"/>	Text
Task Actions	<input checked="" type="checkbox"/>	Text

Output Fields

Name	Type
LastApprover	Text
LastAction	Text

## 入力フィールド

MDM ヒューマンタスクのサービスステップで、次の入力フィールドを設定します。

### テンプレート

必須。開始ステップの【入力フィールド】タブに追加したテンプレートフィールド。

### タスクタイプ

必須。開始ステップの【入力フィールド】タブに追加した taskTypes フィールド。

### 変更リスト

必須。開始ステップの【入力フィールド】タブに追加した入力フィールド。

### タスク名

必須。【全般】タブで指定した、サービスステップの一意の名前。

### タスクアクション

必須。ユーザーがこのステップで実行できるアクションのリスト。区切り文字としてセミコロンを使用します。例えば、変換のための送信;オリジネータへの返送;リリースのようにします。アクションテキストを使用して、次のプロセスステップに関する詳細情報を指定することもできます。各タスクアクションから返される出力は、プロセス内で処理する必要があります。

## 出力フィールド

【最終アクション】フィールドと【最終承認者】フィールドを使用して、割り当てステップや決定ステップなどの追加のプロセスステップを定義し、データやサービスにアクセスしたり、関連するオーケストレーションアクティビティを実行したりすることができます。各プロセスステップは、最終承認者とその承認者が実行した最終アクションに関する情報を返します。

次のフィールドには、追加のプロセスステップから返されたステップ値が格納されます。

### LastAction

変換のための送信、オリジネータへの返送、リリースなどの最終タスクアクション。

### LastApprover

最終アクションを実行したユーザーの ID。

ヒューマンタスクを使用してプロセスを作成するための順を追った手順の説明については、[Defining Cloud Application Integration processes for user-triggered business events](#) を参照してください。

## シェルコマンドの実行

シェルコマンドの実行アクションをサービスステップで使用して、アプリケーション統合の一環としてシェルコマンドへの呼び出しを自動化できます。例えば、このアクションをサブプロセスに追加し、cp または cat のシェルコマンドを実行して、エージェント上のファイルを移動したり、特定ディレクトリのファイルコンテンツを読み込んだりします。

**注:** プロセスでこのアクションを使用する場合、シェルコマンドを実行する Secure Agent を、アプリケーション統合コンソールの **[サーバーの環境設定]** > **[システムサービス]** リストで必ず選択します。ただし、このサービスステップを含むサブプロセスでシェルコマンドを呼び出すと、Cloud Server で実行中のプロセスから、シェルコマンドにアクセスできます。このステップでのシステムコマンドの実行は、推奨されません。

### シェルコマンドサービスの有効化

このアクションを含むプロセスを実行する前に、アプリケーション統合コンソールで、次の手順を使用してシェルサービスを有効にします。

1. アプリケーション統合コンソールで、**[サーバーの環境設定]** を選択します。
2. **[サーバーの環境設定]** リストから、シェルサービスを実行するエージェントを選択します。
3. **[システムサービス]** タブで、**[シェルサービス]** を選択します。
4. **[保存]** をクリックします。

### 入力フィールドと出力フィールド

シェルコマンドの実行用にサービスステップを含めるときに、次のオプションを使用できます。

	名前	タイプ	説明
入力フィールド	コマンド	テキスト	必須。実行するコマンドの名前。これはオペレーティングシステム (Windows または Linux) に固有です。
	文字セット	テキスト	実行するコマンドの文字セット。指定されない場合のデフォルトは、UTF-8 です。
	マージの出力	チェックボックス	チェックすると、出力のコンテンツがエラー文字列とマージされます (ある場合)。デフォルトでは、これは有効になっています。
	入力	テキスト	入力に使用するファイル名とパス (ある場合)。

	名前	タイプ	説明
	作業ディレクトリ	テキスト	必須。コマンドが実行されるディレクトリのパス。
出力フィールド	エラー文字列	テキスト	返されたエラーの文字列（ある場合）。
	応答コード	テキスト	0（成功）または1（失敗）。
	出力文字列	テキスト	返された出力の文字列（ある場合）。

次の図は、シェルコマンドの実行に必要なプロパティを示します。

Run a Shell Command Properties

General

Service

Input Fields

Fault Handling

Timer Events

Message Events

Service Type: System Service

Action: Run a Shell Command

Description: Automated step that executes a shell command

Input Fields

Name	Required	Type	Description
Command	<input checked="" type="checkbox"/>	Text	Command name to be executed
Character Set	<input type="checkbox"/>	Text	Character set of the command to be run.
Merge Output	<input type="checkbox"/>	Checkbox	Check to merge the contents of the output with the error string, if any.
Input	<input type="checkbox"/>	Text	File name and path to use for input, if any.
Working Directory	<input checked="" type="checkbox"/>	Text	Path of the directory where the command executes.

Output Fields

Name	Type
Error String	Text
Response Code	Text
Output String	Text

## クラウドタスクの実行

サービスステップを使用して、データ統合タスクを実行できます。サービスステップを通じて、マッピングタスクと同期タスクを実行できます。

次の手順を実行して、サービスステップを通じてデータ統合タスクを実行します。

1. プロセスにサービスステップを追加します。
2. **【サービス】** タブで、サービスタイプを **【システムサービス】** として選択して、アクションを **【クラウドタスクの実行】** として選択します。
3. **【入力フィールド】** タブで、次の必須プロパティを設定します。
  - **【タスク名】** : Process Designer で実行するデータ統合タスクの名前。マッピングタスクと同期タスクのリストから選択できます。キーワードを入力して、マッピングタスクまたは同期タスクを検索することもできます。
  - **【タスクが完了するまで待機する】** : Process Designer が実行を再開する前に、タスクの完了を待機するかどうかを定義します。
  - **【最大待機】** : Process Designer がタスクの完了を待機する最大秒数を定義します。デフォルト値は 300 秒です。

**重要:** [最大待機] の値は 604800 秒、つまり 7 日間よりも短く入力する必要があります。Process Designer は、タスクの完了を最大で 7 日間待機します。

- [クラウドタスクエラー時に終了] : マッピングタスクまたは同期タスクが失敗したときにプロセスがフォールトするかどうかを定義します。

次の図は、クラウドタスクの実行サービスステップに対して設定できる入力フィールドを示します。

Name	Required	Type	Description
Task Name	<input checked="" type="checkbox"/>	Cloud Task	The name of the task to start.
Wait for Task to Complete	<input type="checkbox"/>	Checkbox	Wait for the task to complete before returning.
Max Wait	<input type="checkbox"/>	Integer	The maximum time in seconds to wait for a task to complete.
Fail on Cloud Task Errors	<input type="checkbox"/>	Checkbox	Throw fault if ICS task completed with errors.

Name	Type
Run Id	Integer
Task Status	Text
Success Source Rows	Integer
Failed Source Rows	Integer
Success Target Rows	Integer
Failed Target Rows	Integer
Start Time	Date Time
End Time	Date Time
Error Message	Text

クラウドタスクの実行ステップが含まれるプロセスをパブリッシュして実行したら、タスク実行履歴をアプリケーション統合コンソールで表示できます。

## フォールト処理

アプリケーションを Process Designer と統合すると、エラー状態を次のように処理できます。

- 境界イベントを使用してプロセスのフォールトをキャッチする。
- 出力フィールドを使用するか、フォールトをスローするためにプロセスにスローステップを追加して、プロセスからフォールトを返す。

## フォールト処理と境界イベント

境界イベントは、特定のステップの境界、つまりイベントが定義されているステップの範囲内で発生するエラーをキャッチするイベントです。プロセス内のステップのフォールト処理を有効にした場合、境界イベントを定義していることとなります。フォールトハンドラは、プロセスが実行されると、そのステップ上のフォールトをリスンします。フォールトがキャッチされるとステップは中断され、プロセスは、フォールト処理用に定義されたパスに従います。フォールトパスで行える処理の例としては、電子メール通知を送信してプロセスを終了する、指定された間隔を待機した後再試行する、プロセスを中断する、別のパスを使用してプロセスを完了する、などがあります。

例えば、ホストシステムにアクセスできない、または無効なデータがプロセスで提供されたなどの理由で、REST サービスがプロセスにエラーコードを返す場合に発生するフォールトをキャッチできます。

**注:** ステップのフォールトハンドラは、中断する境界イベントです。これは、フォールトが発生したときに、プロセスはフォールトパスに従い、ステップのメインパスは実行されないことを意味します。

## フォールトハンドリングのガイドラインのデザイン

フォールトハンドラを定義するときに、次のポイントに注意する必要があります。

- 境界イベントを、1つのステップで発生させるか、一連のステップで発生させるかを判断します。境界イベントが一連のステップで発生する場合は、次の2つの選択肢があります。
  1. ステップごとに個別にフォールトハンドリングを追加します。
  2. 1つの包括的なプロセスの内部に複数のステップを定義します。  
後者のオプションでは、サブプロセスステップにフォールトハンドラを追加し、ラップされたプロセスを起動できます。
- フォールト処理用の並列パスステップがある場合、フォールト情報が含まれる各フィールドに、一意のフィールド名を付けるようにします。フォールト情報の出力フィールドはプロセスレベルで定義するため、あるパスのフォールト情報が別のパスのフォールト情報を上書きする可能性があります。これを回避するために、並列パスステップの各パスで、フォールト情報に一意のフィールド名を設定するようにします。
- フォールトを処理するには、いくつかのオプションがあります。
  1. キャッチされていないフォールトがある場合、プロセスを中断して、アプリケーション統合コンソールを使用してデバッグします。この場合、プロセスの詳細プロパティで [キャッチされないフォールト時に中断:] を選択する必要があります。フォールトをキャッチしないと、中断されてアプリケーション統合コンソールに表示されます。フォールトをキャッチした後で、再スローすることもできます。
  2. フォールトを記録して (例: 電子メール通知の送信) 続行します。
  3. 待機ステップを追加して一定の期間待機し、ジャンプステップを追加してステップを再試行します。ただし、フォールトパスからジャンプステップを使用することはできません。これは、メインプロセスのパスにマージして戻ってからジャンプステップを追加する必要があることを意味します。
- フォールトの詳細 (faultInfo プロセスオブジェクト) は任意のタイプ値として取得されるため、サービスから返されるフォールト情報の構造がわかっている場合は、XQuery を使用して詳細にアクセスできます。

フォールトは、次のものによってトリガされる可能性があります。

- サービスステップ (サービスコネクタまたはプロセスで作成された自動ステップを使用)。
- 作成ステップ (Process Designer で使用できる特別な種類のサービスステップ)。
- ホストシステムのインタラクション。
- プロセスで定義されたスローステップ。
- その他のフォールト。

## サービスコネクタによってトリガされたフォールト

フォールトがサービスコネクタによってトリガされた場合、faultInfo プロセスオブジェクトにエラーの説明が含められ、これがプロセスに戻されます。

Process Designer でフォールト処理を有効にすると、SOA コネクタが生成した可能性のある明示的な runtimeError フォールトもキャッチする可能性があります。以下に例を示します。

```
<wsdl:operation name="read">
  <wsdl:input message="tns:readRequest"></wsdl:input>
  <wsdl:output message="tns:readResponse"></wsdl:output>
  <wsdl:fault name="runtimeError" message="tns:faultResponse"/>
</wsdl:operation>
```

## フォールト処理をサポートするステップのタイプ

次のステップタイプでフォールト処理を有効にできます。

- 作成
- サービス
- サブプロセス

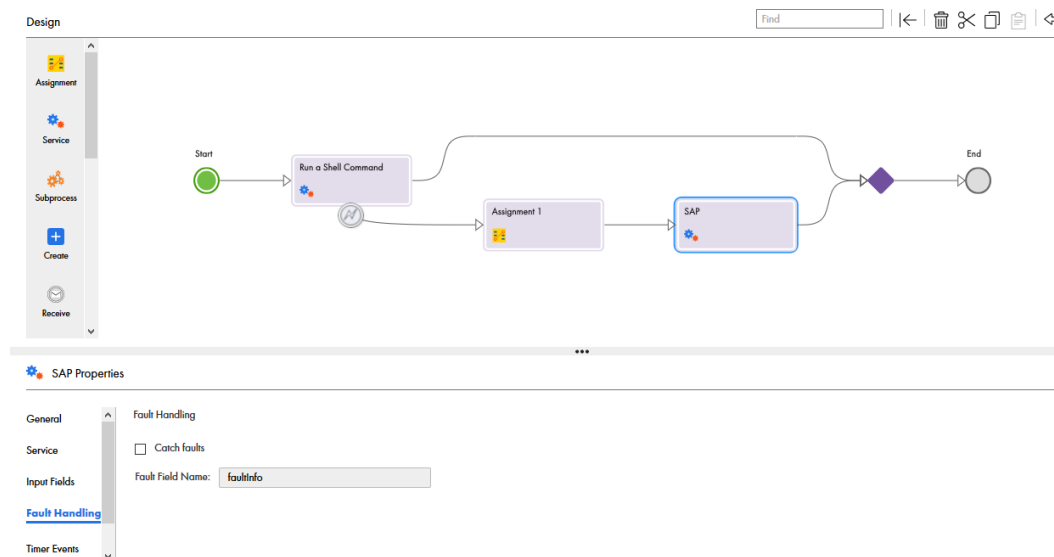


有効にすると、フォールトハンドラは個別のステップ上のすべてのフォールトをキャッチします。これにより中断が発生するため、プロセス内に個別の実行パスを作成します。フォールト名フィールドを指定すると、プロセスオブジェクトのすべてのフォールトの詳細を取得できます。

これらの各ステップタイプの [フォールト処理] タブで、これらのプロパティを設定できます。

プロパティ	説明
フォールトの取得	必須。このステップでフォールト処理を有効にする場合に選択します。デフォルトで、このオプションは無効に設定されています。
フォールトフィールド名:	[フォールトの取得] が有効な場合は必須。フォールト名を表します。デフォルトは <code>faultInfo</code> です。

次の図は、フォールト処理が有効なサービスステップを示します。



スローステップで、特定のフォールトタイプ、理由、および説明を設定することもできます。フォールト情報を渡すために出力フィールドを使用する代わりに、パターンまたはプロセスによってキャッチされたときに、他のプロセスからスローステップを呼び出すことができます。

## プロセスからフォールトを返す方法

1. **出力の使用。**これは、エラーが発生する可能性があることがわかっており、データ決定ステップを使用して通常のプロセスフローに基づいてプロセスを分岐する場合に適しています。
2. **フォールトのスロー。**これは、エラーへの適切な応答が、エラーを呼び出し元に渡すか、エラーをログに記録してプロセスを中断することのみである場合に適しています。例えば、HTTP ステータスコード 403 Forbidden で応答する Web サービスへのアクセスを試みた場合、これは、アクセスが拒否されたことが理由で、サービスは要求を理解したがアクションの実行は拒否していることを示します。多くの場合、このエラーは通常のプロセスフローでは解決しない可能性が高いため、フォールトをスローすることが、これを処理する最良の方法です。

### ステップの使用状況のスロー

プロセスにスローステップを追加すると、スローステップによって、エラー情報がプロパゲートされます。これにより、例えばサブプロセスステップなどを介して、プロセスの呼び出し時にフォールトを捕捉できます。

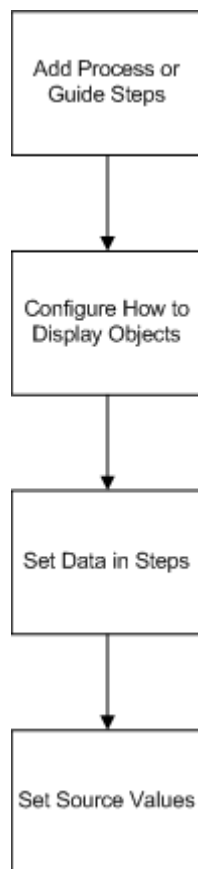
以下の点に注意してください。

1. 名前は文字列値である必要があります。
2. スローは終了ステップであるため、送信リンクはサポートしません。
3. スローステップは、フォールトハンドラを使用して捕捉された情報を再スローするために使用できます。その場合、ステップのペイロードは、サービス、サービスコネクタ、または SOA コネクタによって返されるパラメータを格納した `faultInfo` 要素です。
4. スローステップを使用して、プロセスの一部として使用する特定のフォールト情報を構成することもできます。

## 第 3 章

# データの表示

次の図は、この章で学ぶ内容を示しています。



## オブジェクトの選択と表示

1つのステップ内で、ユーザーが1つ以上のオブジェクトを選択するように設定する必要がある場合があります。この情報は、次のようにさまざまな方法で表示できます。

- オブジェクトを名前で検索するためのポップアップ検索ダイアログ。
- 1つのオブジェクトを選択できる選択リスト。
- 複数選択のための複数選択リスト。

- オブジェクトのテーブル。オブジェクトの中で、別のカラムに表示するオブジェクトのフィールドを選択できます。  
1つのオブジェクトしか選択できない場合、Guide Designer はラジオボタンを含むカラムを追加します。それ以外の場合、チェックボックスを追加します。

**注:** 1つのオブジェクトのみを表示する場合、オブジェクトのフィールドを1行の複数カラムのテーブルとして表示できます。

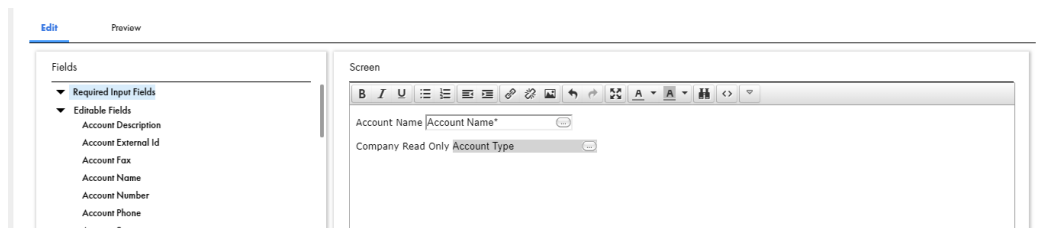
デフォルトでは、ユーザーが更新可能なフィールドの情報を検索するときに、Guide Designer は検索ダイアログボックスを表示します。デフォルトを無効にして、選択リストまたはテーブルを使用してオブジェクト情報を表示するように、Guide Designer を指定できます。選択リストとテーブルは、表示する目的のオブジェクトが出現する可能性がある位置までフィルタリングするクエリを使用して作成されます。例えば、Guide Designer に、アカウントに関連付けられている連絡先のみを表示できます。

各オブジェクトに表示されるフィールドはカスタマイズできます。これらのオブジェクトから1つのフィールドのみが表示されている場合、Guide Designer はこの情報をリストとして表示します。ただし、リードの姓名など、複数のフィールドを表示している場合、Guide Designer は、テーブル内で各フィールドをそれぞれのカラム内に表示します。

**注:** 更新するフィールドが複数のオブジェクトタイプを参照する可能性がある場合（つまり、**[フィールドプロパティ]** ダイアログ内の**[参照]** コントロールに、複数のオブジェクトタイプが含まれている場合）、オブジェクトクエリは使用できません。この状況に該当する場合（通常のクエリを使用できない場合）、詳細クエリの使用を検討する必要があります。

## 選択リストを使用したフィールドの挿入

フィールド名を追加または使用する場所はたくさんあります。



ユーザーがガイドを実行すると、Process Designer は、フィールドをオブジェクト内の値に置き換えます。例えば、**[口座の種類]** フィールドが挿入されて、**[口座の種類]** が **[グレード 1]** の場合、ユーザーがガイドを実行しているとき、フィールド名の代わりに **[グレード 1]** が表示されます。

ステップには、3種類のリストがあります。

- **読み取り専用フィールド:** 読み取り専用フィールドは、値を変更できないフィールドです。このフィールドはグレー表示されるため、読み取り専用であることがわかります。  
フィールドの前のテキストは、フィールドの前の説明テキストとして追加されたものです。説明を追加していなければ、ユーザーには、何の情報が表示されていたかがわかりません。

**注意:** オブジェクト ID がキャンパス内に読み取り専用フィールドとして配置されるとき、またはテーブル内のカラムに配置されるときに、オブジェクト ID はオブジェクトへのリンクとして表示されます。オブジェクト ID の値を表示する場合は、テキストフィールドに「**{!Id}**」として入力します。

- **編集可能フィールド:** 編集可能フィールドを使用すると、ユーザーは、フィールドの現在の値を表示だけでなく、その値を変更できます。

- 

**[...]** ボタンをクリックして、フィールドのプロパティダイアログを表示します。詳細については、[「データ型とフィールドプロパティの概要」 \(ページ 10\)](#)を参照してください。

## 自動生成フィールド

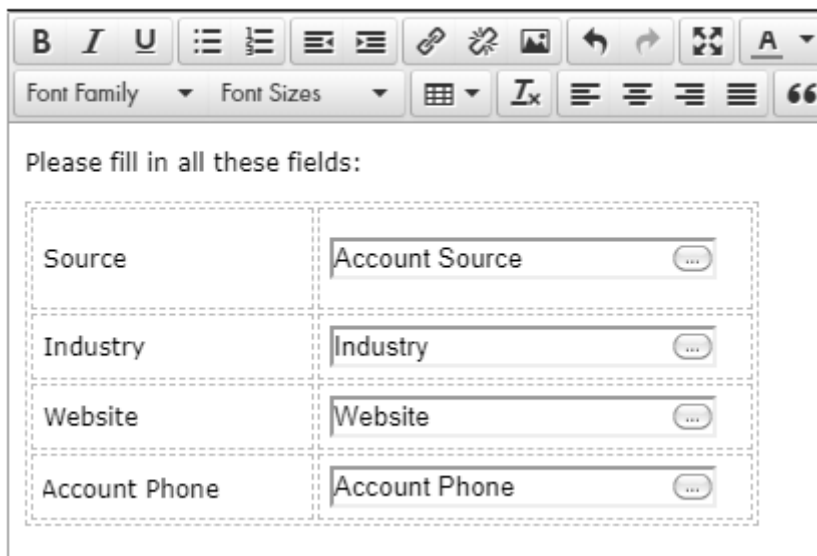
フィールドは仮想にできます。つまり、実際は複合キーまたは複合フィールドのフィールドをオブジェクトに設定できます。Guide Designer は、この仮想の自動生成されたフィールドの名前を表示します。その際、「Object\_name」 + 「\_IID」の形式を使用します。

## テーブルのフィールドの挿入

ステップにフィールドを挿入するときに、選択リストのフィールド名をクリックするとラベルとテキストフィールドが挿入されます。ここに、フィールド情報が表示されます。複数のフィールドを挿入する場合、ステップが読みづらくなり、見栄えが悪くなる可能性があります。テーブル内に要素を配置することで、これら両方の問題を解決できます。

次に例を示します。

Screen



テーブル内のフィールドを処理する際は、次の点に注意します。

- カーソルが右端のセルにない場合、Guide Designer は、現在のカーソル位置にラベルを挿入します。フィールドは、カーソルの右側のセルに挿入されます。右側のセル内にデータがある場合、Guide Designer はこの情報にフィールドを付加します。
- カーソルが右端のセルにあり、セルが空白でない場合、Guide Designer はラベルとフィールドを、後続の行の先頭にある2つのカラムに挿入します。その行にテキストが含まれる場合、Guide Designer は、現在の行のすぐ下に行を挿入して、ラベルとフィールドを挿入します。その行が空の場合は、そのセルに情報を配置します。
- カーソルが空白の右端のセルにあるか、テーブルにカラムが1つしかない場合、Guide Designer は（ラベルではなく）フィールドをこのセルに配置します。

## フィールドのテキストとしての入力

次のようなテキストボックスに情報を入力するときに、テキストとフィールドの値を入力できます。

フィールド値を指定するには、コンテンツをソースとして選択し、テキストボックスの右側の小さいアイコンをクリックします。表示されるフィールドのリストから、入力するフィールドを選択します。フィールドの名前がわかっている場合は、その名前をテキストボックスに `{ $name_of_field }` として入力することもできます。

この例に、現在のオブジェクトからの4つのフィールドの値を、[説明] フィールドの [コンテンツ] として示します。

The screenshot shows the 'Event Properties' configuration interface. It features a sidebar with 'General', 'Create', and 'Input Fields' sections. The main area is titled 'Input Fields (2)' and contains a table with columns for 'Name', 'Required', and 'Value'. The 'Start Date Time' row has a 'Required' checkbox and a 'Value' field with a dropdown menu set to 'Specific date' and a time selector set to '12:00 PM'. The 'Description' row has a 'Required' checkbox and a 'Value' field with a dropdown menu set to 'Content'. A tooltip is visible over the 'Content' dropdown, listing available fields: '\$Account.Name', '\$Who.Lead.Id', and '\$What.Campaign.Id'. There are also 'Edit Screen' and 'Preview Screen' buttons at the top right, and an 'Add Field' button at the bottom right.

フィールド値と一緒に表示するために、その他のテキストを追加できます。例えば、次のように指定できます。

名前: `{ $name }`

## カラムの表示

**【表示フィールド】** フィールドを使用して、テーブルにオブジェクトのどのフィールドを表示するかを Guide Designer に対して指定します。フィールドが更新可能であると、ユーザーが1つのオブジェクトのみを選択できる場合はラジオボタン、ユーザーが複数選択できる場合はチェックボックスのいずれかが各行にも表示されます。

カラムを追加するには、**フィールドカラムの [追加]** ボタンを使用します。名前の右側の矢印ボタンを使用して、カラムの順序を変更できます。デフォルトのカラムヘッダーを使用しない場合、任意のテキストを入力します。

ここに4カラムのテーブルがあります。

Field Properties

Field:	Account_ID (input)		
Reference To:	Account		
Required:	<input type="checkbox"/>		
Default Value:	Content	--None Selected--	
Display Fields:	<b>Field</b>	<b>Label</b>	<b>Properties</b>
	Account Name	Account Name	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
	Billing City	Billing City	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
	Billing State/Province	Billing State/Province	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
	Account Phone	Account Phone	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
Add Field			
<input checked="" type="checkbox"/> Paginate Results <input type="checkbox"/> Allow Filtering <input checked="" type="checkbox"/> Allow Column Sorting			
Hover text:			
Height (rows):			
Show List:	None		

OK Cancel

3つのチェックボックスは次のとおりです。

- ページネーションの結果: 選択した場合、出力にページ数が付けられます。
- フィルタリングを許可: 選択した場合、ユーザーは表示結果の一部を選択するために、テキストボックス内にテキストを入力できます。
- カラムソートを許可: 選択した場合、ユーザーはカラムのコンテンツを基準に結果をソートするために、カラムヘッダーをクリックできます。

注: ID フィールドが [ルックアップ] ダイアログからの結果に含まれている場合、ID の値が表示されます。

## ステップでのデータの設定

作成するステップによっては、そのステップのアクションの発生前に、ユーザーに情報の入力を求めるものがあります。このような例には、サービスステップや作成ステップがあります。これらとは異なり、ステップの実行開始時には、そのステップに必要な情報をすでに持っているものがあります。ステップに必要な情報がある場合、アクションは自動的に発生し、ユーザーがその情報を見ることはありません。その代わりに、ユーザーは後続のステップを見ることとなります。このトピックでは、例として作成ステップを使用しますが、その他の種類のステップでもデータの設定方法に違いはありません。

ステップのプロパティセクションで、アクションまたはオブジェクトと関連付けられたフィールドに関する情報を参照できます。エンティティまたはオブジェクトが選択されない場合、[情報] タブ表示されるデータはありません。

Name	Required	Type	Description
Account ID	<input type="checkbox"/>	Object ID (Salesforce:Account)	
Private	<input type="checkbox"/>	Checkbox	
Name	<input checked="" type="checkbox"/>	Text	
Description	<input type="checkbox"/>	Text Area	
Stage	<input checked="" type="checkbox"/>	Picklist	
Amount	<input type="checkbox"/>	Currency	
Probability (%)	<input type="checkbox"/>	Percent	
Quantity	<input type="checkbox"/>	Number	
Close Date	<input checked="" type="checkbox"/>	Date	
Opportunity Type	<input type="checkbox"/>	Picklist	
Next Step	<input type="checkbox"/>	Text	
Lead Source	<input type="checkbox"/>	Picklist	

このセクションには、このステップがオポチュニティオブジェクトを作成するときにデータを提供できるすべてのフィールドの一覧が表示されます。実際に使用するフィールドは、[入力] タブで命名され、設定されます。

## ステップでのフィールドの使用

プロパティシートで指定するデータ型には、数値、テキスト、時間などがあります。これらのデータ型によって、データがプロセス内で処理され、ガイド内に表示される方法が決まります。

次の図に、[アカウントタイプ] フィールドを示します。

Field:	Account Type
Required:	<input checked="" type="checkbox"/>
Default Value:	Content ▼ Partner ▼
Hover text:	

Show List: None ▼

[デフォルト値] フィールドと [リストを表示] フィールドは、ステップでフィールドを定義するときのみ表示されます。

- **デフォルト値:** [コンテンツ]、[フィールド]、または [計算式] を選択します。
  - コンテンツには、特定の値を入力できるコントロールが表示されます。
  - フィールドには、デフォルト値を選択できるリストが表示されます。
  - 計算式には、ここに入力する計算式を使用して値を設定できます。

**注:** 更新用画面にあるフィールドにデフォルト値を選択すると、フィールドの現在の値がステップの表示前に設定されていない限り、デフォルト値はユーザーに表示されません。設定されている場合、ユーザーにこの現在の値が表示されます。既存の値がないためにデフォルト値が常に表示されるので、サービス呼び出しのステップは異なります。


- **リストを表示:** 詳細は、「[オブジェクトの選択と表示](#)」(ページ 99)を参照してください。



# ソース値の設定

フィールドに割り当てられる値のソースを Guide Designer に対して指定できます。次の図に、フィールドが異なるソースを持つ作成ステップの一部を示します。

Input Fields (2)

Name	Required	Value
Account Name	<input checked="" type="checkbox"/>	Content <input type="text"/> 
Parent Account ID	<input type="checkbox"/>	Field <input type="text"/> Account ID <input type="text"/>

- **追加:** 入力したフィールドのタイプが [ObjectList] の場合、この項目を選択すると、そのリストに項目を追加できます。
- **コンテンツ:** ステップが使用する値は、ここで定義するもので、変更されることはありません。例えば、3つの値のいずれかを持つと定義された [評価] フィールドがある場合、このガイドを設計する際にこれらの3つのいずれかの値に設定されます。ここで設定する値は、このオブジェクトの作成時に使用される値です。ただし、これがテキストの場合、テキストには組み込みフィールドも含まれる場合があります。この例には、3種類の異なるコンテンツフィールドがあります。Process Designer が設定されるデータの種類を認識しているため、この違いがあります。例えば、[変換] がチェックボックスとして定義されており、[評価] が事前定義済みの値を持つ選択リストとして定義されており、[郵便番号] がテキストとして定義されるなどです。
- **フィールド:** 値はオブジェクト内のフィールドから取得されます。
- **計算式:** 次のような数値演算子を使用します。
  - +
  - -
  - \*
  - div
  - mod組み込み関数と XQuery 関数を使用して、値を設定できます。
- **クエリ:** (これは表示されません)。Guide Designer に、配置する情報を指定します。例えば、Guide Designer で「Acme」という名前の会社を検索します。
- **画面:** (ガイドのみ) [入力画面] タブの編集領域内に入力フィールドを挿入する必要があります。

## ソース値の設定: コンテンツ

コンテンツソースは、ガイドの設計者が、ステップへの入力として渡される値を決定するものです。例えば、コンテンツを使用して、ガイドのユーザーに説明の入力を求めることなく、「製品のデモンストレーション」にイベントを作成する作成ステップの説明フィールドを設定します。

場合によっては、テキストとフィールドの値を含むコンテンツを作成する必要があることがあります。例えば、人物の名前と住所を含む「郵送」フィールドを作成するとします。この情報は、フィールド文字列を含めることで追加できます。

フィールドが複数の場合、[x] アイコンをクリックし、リストを表示します。

フィールドの選択後、Process Designer はカーソルの位置にあるテキストボックスにフィールドコードを配置します。必要に応じていくつのフィールドでも個別に選択できますが、一度に選択できるのは1つのフィールドのみです。

次に例を示します。

Name	Required	Value
Start Date Time	<input type="checkbox"/>	Specific date   12:00 PM
Description	<input type="checkbox"/>	Content

この挿入されたフィールドインジケータに、テキストを追加することもできます。例えば、`{Account.name}`の前に、「Name: 」と入力できます。

Process Designerがこのステップを実行するときに、フィールドインジケータを、フィールドから抽出されたテキストで置き換えます。

## ソース値の設定: フィールド

[ソース] に [フィールド] を選択すると、Process Designerは縮小されたリストを右側に表示します。このステップの実行時に、Process Designerがオブジェクトに書き込む値は、その値を持つフィールドを選択することで選択できます。フィールドを選択するその他の場所と同様に、選択リストの下部には、フィールドを使用できる関連オブジェクトが表示されることがあります。

Name	Required	Value
Account Name	<input checked="" type="checkbox"/>	Field   Click To Select Field
Parent Account ID	<input type="checkbox"/>	Field   Click To Select Field
Annual Revenue	<input type="checkbox"/>	Field   Parent Account > Annual Revenue

## ソース値の設定: 計算式

値を設定するために計算式を使用できる、数値入力フィールドを取得するフィールドです。例えば、これを割り当てステップと一緒に使用して、既存のフィールドに数値を追加できます。数値演算子 (+、-、\*、div、mod など) を使用して、フィールドの演算を実行します。

計算式内に任意の XQuery 関数を使用することもできます (これらは <http://www.w3.org/TR/xpath-functions-30/> で定義します)。

### 評価メモ

Process Designer は、予想どおりに式を評価します。以下の点に注意してください。

- `$Field1.Field2.Field3` は、Field3 のコンテンツを参照解除するために前処理され、コンテンツをその値にバインドします。
- フィールド名の特殊文字は削除されます。数字で始まるフィールドにはプレフィックス「x」が付けられません。
- `$Account` が値渡しオブジェクトである場合、そのコンテンツを取得するには、XPath パス式を使用する必要があります。例えば `$Account/Owner/Name` です。
- オブジェクトリストは、ID 値 (参照渡しオブジェクトリスト用) またはレコードの XML 表現のいずれかを含むシーケンスである必要があります。例えば、`($a,$b)` という式を使用して、2 つのオブジェクトリストを結合できます。
- リスト内のオブジェクト参照は、`(a; b; c)` のようにセミコロンで区切られます。

使用可能な関数の詳細については、「[関数の使用](#)」(ページ 29)を参照してください。

## ソース値の設定: クエリ

クエリを使用して、Where 句で選択されたすべてのオブジェクトからオブジェクト情報を取得します。

クエリコントロールはさまざまなステップやダイアログで表示されるため、表示される内容は若干異なる場合があります。

表示するために実行するのは、SQL WHERE 句を作成することです。テキスト領域に直接条件を入力できますが、**[条件の追加]** ボタンと **[ソート順]** 選択リストをクリックの方が簡単です。Process Designer がこの情報を挿入した後で、必要に応じて編集できます。フィールドの内部名が何かが不明な場合があるため、Process Designer に情報を取得させることをお勧めします。ここで入力されるテキストは、標準 SQL WHERE 句です。また、このテキストには、WHERE キーワードは含まれません。

各条件に、次の 4 つの部分があります。

- オブジェクト内のフィールドの名前。この例では、このフィールドはオブジェクト内に含まれているフィールドの 1 つです。
- 左側のフィールド値と右側の値を比較するときに Process Designer が使用する演算子。
- 比較されるデータの種類。選択肢は **[コンテンツ]**、**[フィールド]**、または **[計算式]** です。**[フィールド]** を選択した場合、現在のオブジェクトの最初のフィールド (**[条件]** ダイアログの左端にあるフィールド) 内のコンテンツと、クエリ対象のオブジェクトのタイプ内のフィールドのコンテンツが比較されます。
- フィールドが左側のデータが、比較対象のデータです。ソースが **[コンテンツ]** の場合、この領域内をクリックすると、右側に小さなアイコンが表示されます。これを選択すると、Process Designer に、比較の実行時に使用される別のオブジェクト内のフィールド名を選択できる選択リストが表示されます。

**[ソート順]** 選択リストボタンを押すと、情報が表示されたときにクエリをソートするフィールドを選択できます。例えば、表示される情報を人物の姓でソートできます。この選択リストからフィールドを再選択することで、キーを追加できます。Order By 句は、テキストボックス内に配置されます。これにより、挿入後に編集できます。Order By 句をここで直接入力することもできます。

**注:** SQL LIMIT 句は、Where 句のテキストボックスに入力することで、クエリに追加できます。例えば、「Limit 200」と入力すると、取得した行数を 200 に制限できます。この句を追加するときは、Where 情報の後ろにスペースを追加するか、Where の後に改行して配置します（「Limit」は大文字、小文字、または大文字と小文字を混在させて記述できます）。デフォルトでは、100 行が返されます。ガイドに 100 を超える行が必要な場合、この句を追加する必要があります。

### WHERE 句での条件の指定

WHERE 句内で条件を定義する機能は、Produce Designer との相互作用方法によって異なります。

#### リレーショナルデータベースとの直接的相互作用

ほとんどのデータベースは、SQL WHERE 演算子を完全にサポートしています。問題が発生した場合は、そのデータベースのマニュアルを参照してください。

#### Informatica コネクタの使用

複数の条件が必要な場合は、単純な同族関係（フィールド演算子の値）のみがサポートされます。AND または OR のいずれかを使用して、複数の条件を結合できます。NOT 演算子を使用して、条件の意味を逆にすることもできます。

#### Salesforce

WHERE 条件は、SOQL 要件に準拠している必要があります。詳細については、Salesforce SOQL のマニュアルを参照してください。

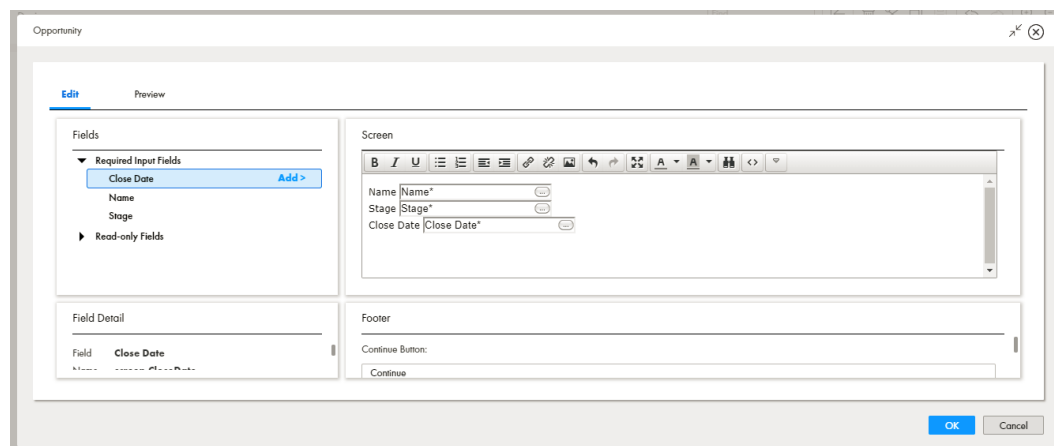
## 回答がないことが予期される場合の WHERE 句の使用

Process Designer では一意と認識されないよくある状況は、複数の条件に基づいて WHERE 句が作成されるときに起こります。このような場合、すべての条件に値があるわけではありません。例えば、アカウントと地域があり、地域は省略可能だとします。Account={\${account}} AND Territory={\${territory}}などの WHERE 句を設定すると、地域 (territory) の値が使用されていない場合は機能しません。

省略可能なフィールドを持つ WHERE 句を作成する場合の解決策は、LIKE 演算子を使用することです。例えば、Account LIKE '%\${account}%' AND Territory LIKE '%\${territory}%'とします。地域が存在しない場合、これは Account LIKE '%\${account}%' AND Territory LIKE '%%'と入力した場合と同じになります。AND の後の LIKE は、常に true と評価されます。

## ソース値の設定: 画面

[ソース] に [画面] を選択すると、このフィールドのデータはエンドユーザーが提供します。つまり、ガイドのユーザーが情報を入力できるように、ステップの [入力画面] タブにフィールドを追加する必要があります。次の例では、3つのフィールドの [入力] タブのソースが [画面] に設定されています。



**注:** 入力するテキストは、テーブルに配置するとより見やすくなります。

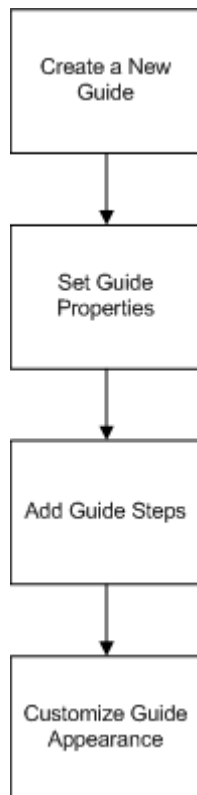
すべてのフィールドと同様に、フィールド内をクリックすると、プロパティダイアログが表示されます。

## 第 4 章

# ガイドのデザイン

ガイドはユーザーにデータのレビュー、入力、または確認を求める画面のセットです。例えば、ステップにアカウントの詳細を表示でき、営業電話のステータスを確認するようにユーザーに指示できます。バックグラウンドでは、ステップはデータを抽出して保存することによって、アプリケーションと連動しています。ガイドは、モバイルアプリまたは PC や Mac などの従来のプラットフォームで動作します。

次の図は、ガイド設計プロセスを示しています。

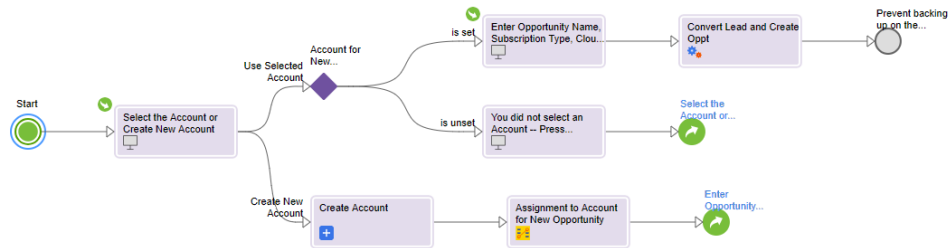


ガイドは専門知識や正式なトレーニングなしで作成できます。

営業マネージャが、新しいアカウントの作成や Salesforce 内の既存のアカウントの更新の際に従うプロセスを自動化するとします。

ガイドを作成するには、次の一般的な手順を実行します。

1. 質問および想定される回答を含む画面を作成します。オプションが定義されると、スクリプトからの一連のステップが、キャンパスのフローチャートとして表示されます。次の図は、ガイドを示しています。

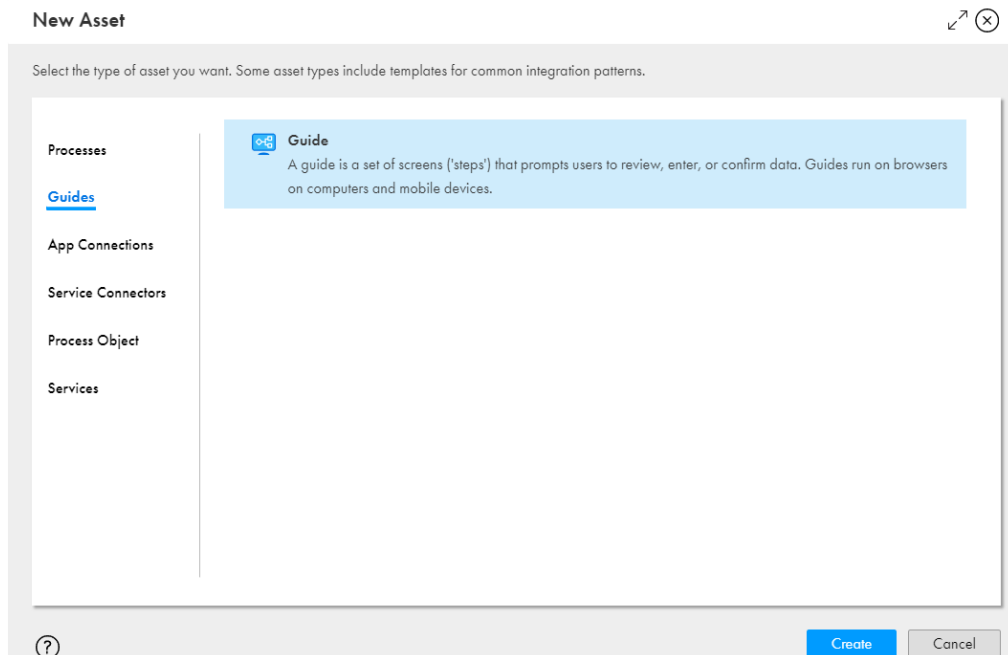


2. オプションの自動化アクションを挿入します。例えば、リードを変換して新しいオポチュニティを作成するプロセスを呼び出します。ガイドを実行するときに、ガイドが適切なパスを経ると、プロセスが呼び出されます。
3. ユーザーが販売アクティビティのためにガイドにアクセスできるように、ガイドをパブリッシュします。

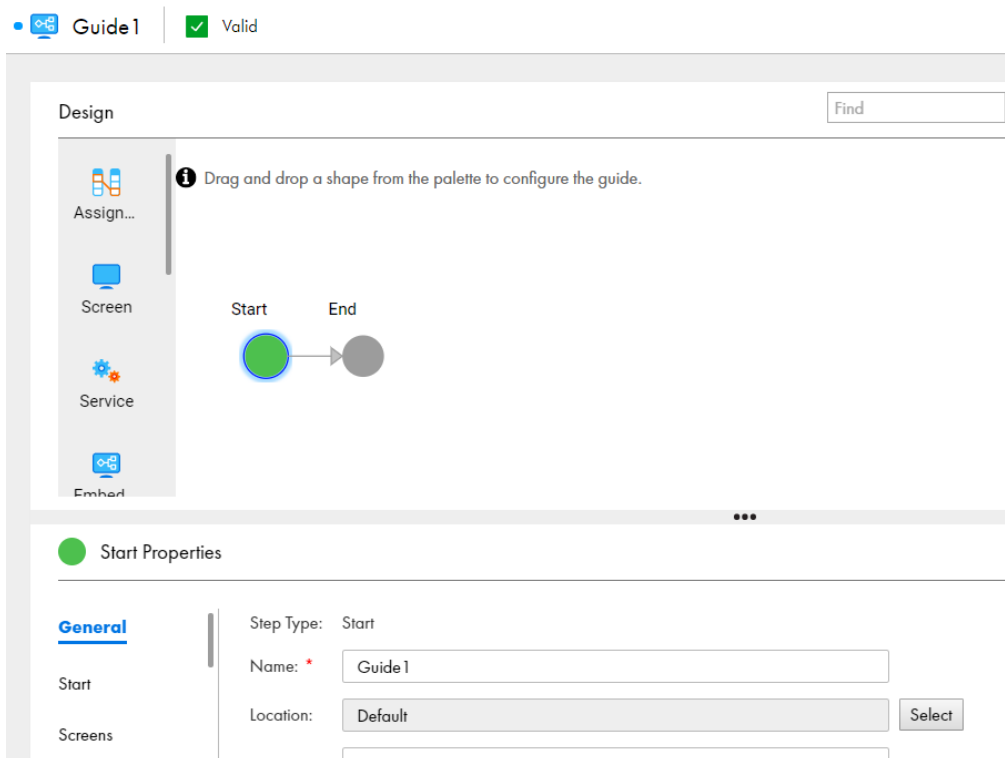
## ガイドの作成

ガイドを作成するには、以下の手順を実行します。

1. アプリケーションの統合で、**[新規]** を選択します。



2. **【新しいアセット】** ダイアログボックスで、**【ガイド】** > **【ガイド】** の順に選択し、**【作成】** をクリックします。  
ガイドのテンプレートが開きます。



3. ガイドのプロパティを入力して、ステップを追加します。

## ガイドのプロパティの設定

1. **【開始】** ステップを選択します。
2. 次のタブにプロパティを入力します。
  - 全般
  - 開始
  - 画面
  - 入力フィールド
  - 出力フィールド
  - 一時フィールド
  - 結果
  - 詳細
  - 付記:

## 全般プロパティ

ガイドには、次の一般的なプロパティを指定できます。

プロパティ	説明
名前	Guide Designer でガイドを識別するためのわかりやすい名前であり、プロセス呼び出しステップなどの他のオブジェクトでガイドを使用できる場合に表示される名前。 名前にはマルチバイト文字を含めることができ、80 文字以内にする必要があります。 パブリッシュされたガイドの名前を変更するには、まずガイドのパブリッシュを解除する必要があります。次に、名前を変更し、ガイドを再パブリッシュします。
場所	プロセスを配置するプロジェクトまたはフォルダの場所。ガイドの場所を選択するには、該当するプロジェクトまたはフォルダを参照するか、デフォルトの場所を使用します。
説明	ガイドの説明。

## 開始プロパティ

ガイド用に次の開始プロパティを定義できます。

プロパティ	説明
適用対象	ガイドに関連付けられているオブジェクトのタイプ。このリストに表示されるオブジェクトは、個別に定義したプロセスオブジェクトまたは定義済みの接続内で生成したプロセスオブジェクトです。リスト内をドリルダウンしてプロセスオブジェクトと接続を表示し、これらの項目の 1 つを選択します。ガイドが別のガイドに組み込まれる場合にのみ、プロセスオブジェクトを選択します。 <b>注:</b> パブリッシュされたガイドの <b>【適用対象】</b> プロパティを変更するには、最初にガイドのパブリッシュを解除する必要があります。次に、 <b>【適用対象】</b> プロパティを変更し、ガイドを再パブリッシュします。 ガイドを特定のオブジェクトに関連付けない場合は、 <b>【任意】</b> を選択します。このオプションを選択すると、ガイドが、オブジェクトのフィールドに自動的にアクセス権限を持つことはありません。ガイドが新しいオブジェクトを作成し、既存のオブジェクト内の情報にアクセスしない場合は、このオブジェクトタイプを使用します。 <b>注:</b> ガイドを実行する場合は、 <b>【適用対象】</b> の設定は <b>【任意】</b> またはサービス内の特定のオブジェクト（プロセスオブジェクト以外）のいずれかである必要があります。
ユーザーを指定して実行 (Salesforce のみ)	Salesforce の場合のみ、ガイドを <b>【現在のユーザー】</b> または <b>【システム】</b> として実行するかどうかを選択できます。 ガイドを実行したユーザーと同じ権限が必要な場合は、 <b>【現在のユーザー】</b> を選択します。 システムレベルのアクセス権限がないとアクセスできないアクションをガイドのユーザーが実行できるようにするためにガイドにシステムレベルのアクセス権限を付与する必要がある場合は、 <b>【システム】</b> を選択します。例えば、ユーザーにアカウントオブジェクトへのアクセス権限がなくても、ガイドが <b>【システム】</b> として実行されていると、ガイドはガイドを実行するために必要な範囲でアカウントオブジェクトにアクセスできます。



## 画面のプロパティ

**【画面】** タブを使用して、ガイドを表示する場所やガイドでサポートする言語を設定します。

**【画面】** ステップ向けに次のプロパティを定義できます。

プロパティ	説明
意図される表示:	ガイドを実行する環境。 次のデバイスのブラウザでガイドを実行する場合は、 <b>【デスクトップ/ターゲット】</b> を選択します。 - デスクトップ - ラップトップ - タブレット  次のモバイルアプリケーションでガイドを実行する場合は、 <b>【スマートフォン】</b> を選択します。 - Informatica - Salesforce S1
複数言語のサポート	ガイドのローカリゼーションでサポートする言語のリスト。 <b>【言語を追加】</b> リストから言語を選択します。

## ガイドの翻訳

**【開始】** > **【画面】** > **【複数言語のサポート】** オプションを使用して、ガイドに表示する言語を選択します。

このセクションでは、以下の項目について説明します。

- [Creating content for a Translated Screen \(ページ 114\)](#)
- [What You Can Change in a Translated Screen \(ページ 116\)](#)
- [Simulating a Translated Screen \(ページ 117\)](#)
- [What Do End Users Do to Select a Translated Guide \(ページ 117\)](#)

**【言語の追加】** を選択して、言語のリストを表示します。

次の図は、**【言語】** リストのサンプルです。

Start Properties

General

Intended Display:

Start

Support multiple languages

Languages (4)

Name	Guide Name	Guide Description
Default	Close Lead	Close a lead in salesforce
Catalan	Close Lead	Close a lead in salesforce
French	Close Lead	Close a lead in salesforce
Spanish	Close Lead	Close a lead in salesforce

デフォルトでは、すべての言語がデフォルト言語の名前と説明を使用します。翻訳された名前と説明を表示するには、各言語に対応する**【ガイド名】**と**【ガイドの説明】**を入力します。

言語を削除するには、**【削除】** をクリックします。

**【デフォルト】** 項目は削除できません。

次の図は、言語が追加されたガイドの【プロパティの詳細】ダイアログボックスを示します。

## Properties Detail for Close Lead

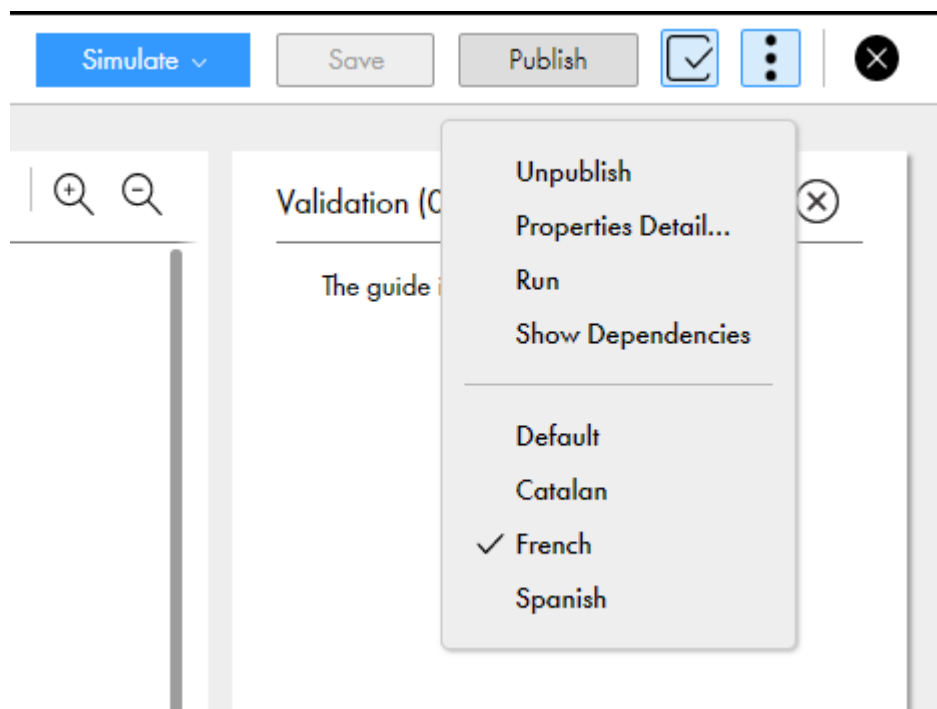
Basic

---

Unique Name:	Close_Lead-1
Publication Status:	<input checked="" type="checkbox"/> Published
Published On:	2018-07-03 15:38
Published By:	sabraham@informatica.com
Applies To:	* Any *
Run As (Salesforce Only):	\$current
Languages:	Default: Catalan(ca), Default: French(fr), Default: Spanish(es)

### 翻訳された画面用コンテンツの作成

次の図に示すように、【アクション】オプションから言語を選択します。

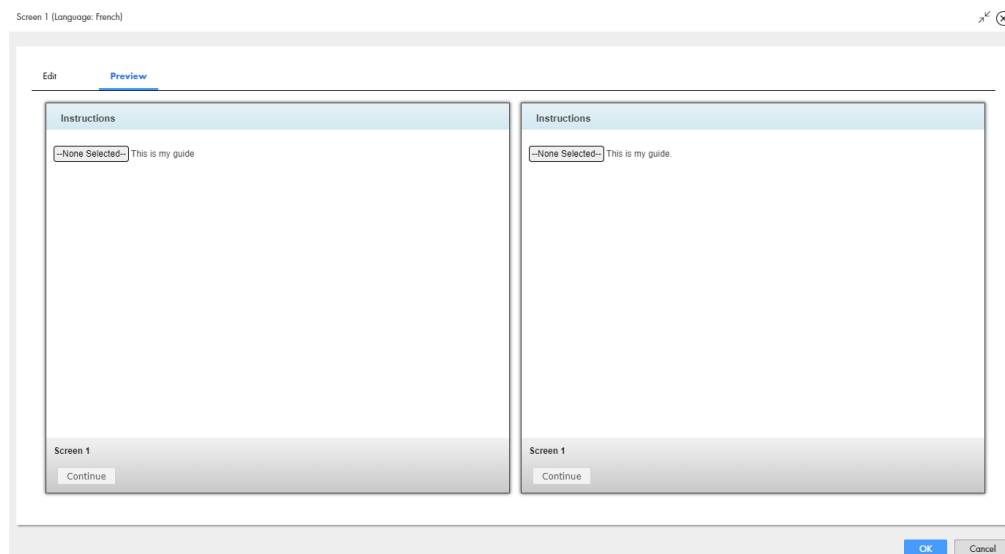


翻訳された画面のコンテンツを編集するには、次の手順を実行します。

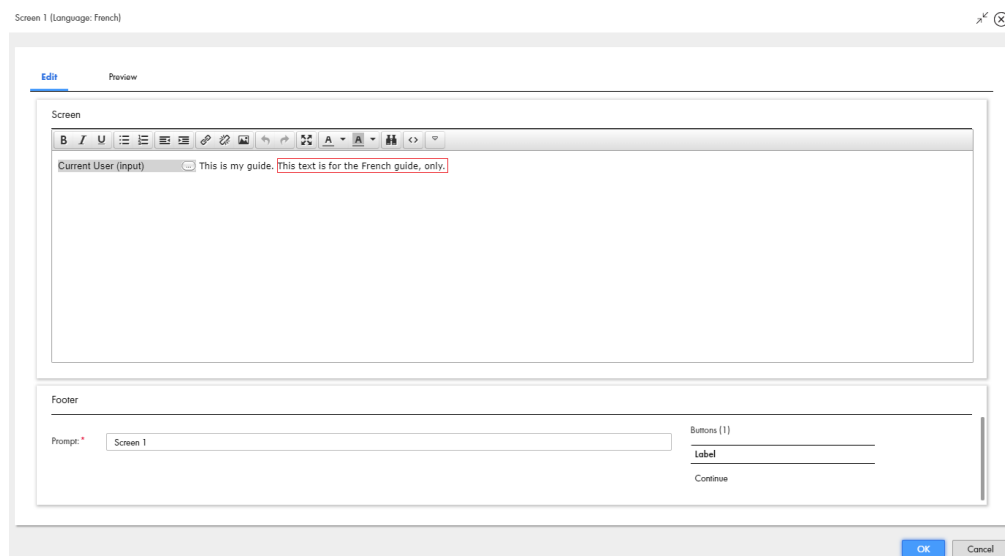
1. **【アクション】** に移動して、言語を選択します。
2. 編集するステップを選択します。
3. ステップのプロパティセクションで **【画面のプレビュー】** をクリックします。  
デフォルト言語用のセクションと、翻訳言語用のセクションが含まれるウィンドウが表示されます。

**注:** デシジョンステップとジャンプステップを除くすべてのステップで、**【画面の編集】** オプションと **【画面のプレビュー】** オプションを使用できます。

次の図では、左側にデフォルト画面、右側に翻訳画面が表示されています。

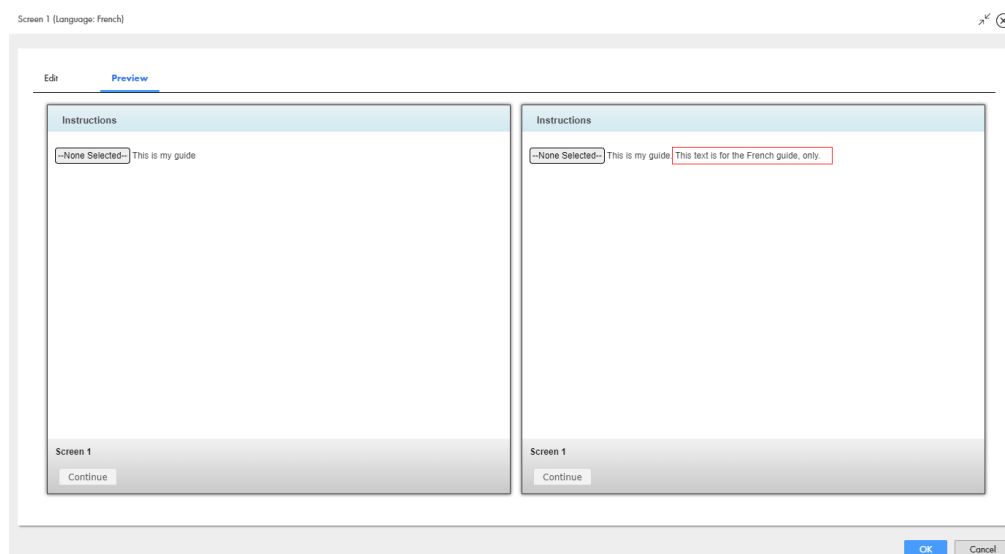


4. **【編集】** タブを選択して、翻訳ガイドに含めるコンテンツを入力します。  
次の図は、コンテンツが入力された **【編集】** タブを示します。



5. **【プレビュー】** タブを選択します。

次の図は、入力したコンテンツが、フランス語の画面にのみ表示される様子を示します。



ガイドのすべての言語に該当する変更を加えるには、[アクション] オプションから **【デフォルト】** を選択し、ステップを選択して **【画面の編集】** をクリックします。

次の表は、デフォルト画面に変更を加えた場合に、翻訳画面がどのようなようになるかを示しています。

デフォルト画面に加えた変更	翻訳画面のアクション
フィールドの追加	最下部にフィールドが追加されます。
フィールドの削除	フィールドが削除されたことを伝えるテキストが、翻訳画面に追加されます。翻訳画面から、この挿入されたテキストを削除する必要があります。
回答の追加	回答が追加されます。回答を各言語用に翻訳するまでは、すべての翻訳でデフォルトの回答が使用されます。
回答の削除	同じ位置にある回答が削除されます。
回答の順序変更	回答の順序が同様に変更されます。
テキストの追加	デフォルトに追加されたテキストが、翻訳されていない言語に追加されます。
テキストの削除	翻訳されていない言語からテキストが削除されます。

### 翻訳画面で変更できるもの

手順、プロンプト、タイトル、および回答の各領域内のテキストのみを変更できます。また、翻訳画面内の情報のレイアウトも変更できます。例えば、翻訳画面内のテーブルに、デフォルト画面の2つの箇条書き項目を配置できます。

ガイドの一部であるテキストは変更できません。例えば、「手順」、「再起動」、「履歴」などです。

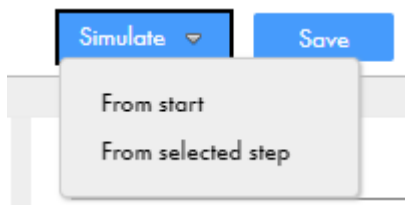
アプリケーションの統合は、内部的にデフォルト画面と翻訳画面を同期し、デフォルト画面とすべての翻訳画面の間の基本的な構造が維持されるようにしています。また、ステップタイプの変更、フィールドの追加、回答の順番変更が行われないようにするため、これらの使用を禁止しています。一部のアクションは、コントロール

ールを選択不可にすることで実行不能になっています。ただし、それでもフィールドや回答を誤って削除してしまう場合があります。フィールドや回答を削除してしまった場合、Guide Designer はエラーボックスを表示し、削除されたものを再度挿入して、この処理を通知します。誤って削除したものに関連付けられていたテキストがある場合は、デフォルトの情報がコピーされます。例えば、回答を削除した場合、Guide Designer は、デフォルトの回答テキストを翻訳画面に挿入します。

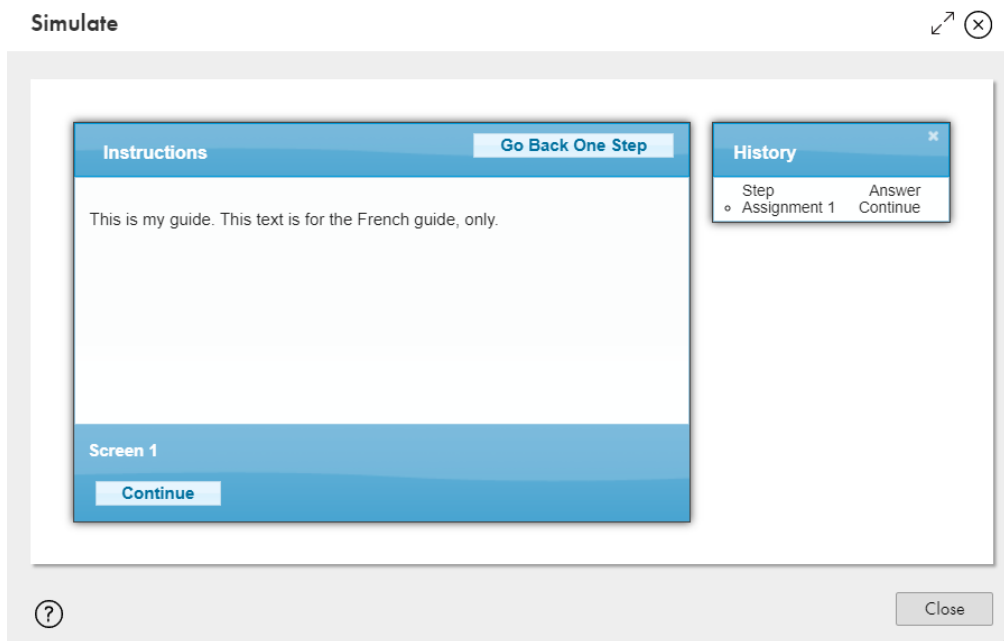
### 翻訳画面のシミュレーション

翻訳画面がユーザーにどのように表示されるかを確認するには、次の手順を実行します。

1. **【アクション】** > **【言語の選択】** の順に移動します。
2. キャンバスで、表示するステップを選択します。
3. ページ上部の **【シミュレート】** リストから、**【選択したステップから】** を選択します。  
次の図は、**【シミュレート】** リストを示しています。



次の図は、シミュレートされたステップを示しています。



### 翻訳されたガイドを選択するためにエンドユーザーがすること

エンドユーザーは、翻訳されたガイドを選択する必要はありません。アプリケーションの統合は、ブラウザのロケールを使用して、表示する言語を選択します。

**Salesforce のみ:** Salesforce ユーザーは、常に 1 つの言語に関連付けられています。使用している Salesforce のバージョンに応じて、**【私の個人情報】** の **【個人情報】** セクション内、または **【私の設定】**、**【個人情報】** の **【言語とタイムゾーン】** 内でこの設定を確認できます。Guide Designer はこの値を確認し、その言語の翻訳ガイドが存在する場合は、そのガイドを表示します。

## 入力フィールドのプロパティ

ガイドに対して使用できる入力フィールドは、ガイドの適用先のオブジェクトに含まれるフィールドです。これは、**【開始】** タブの **【適用対象】** プロパティで指定します。

**【入力】** フィールドの値は、ガイドのすべてのステップ内で使用できます。組み込みガイドは、組み込みガイドからの情報にアクセスする必要がある場合があります。入力フィールドの作成は、情報を組み込みガイドに渡せるように、その情報を定義する方法でもあります。

次の表に、ガイドで使用する入力フィールドカテゴリの名前とタイプを示します。

プロパティ	説明
名前	入力フィールドの名前です。
タイプ	入力フィールドのタイプ。例えば、 <b>【日付】</b> 、 <b>【日付/時刻】</b> 、 <b>【時間】</b> 、 <b>【整数】</b> 、または <b>【テキスト】</b> を選択します。 単純型、カスタムタイプ、または定義された接続タイプの入力フィールドを追加することもできます。入力フィールドを追加するには、 <b>【タイプをさらに表示】</b> を選択し、 <b>【タイプの編集】</b> ダイアログボックスで、 <b>【カテゴリ】</b> に <b>【単純型】</b> 、 <b>【カスタムタイプ】</b> 、または <b>【定義された接続タイプ】</b> を選択します。
説明	入力フィールドの説明です。
必須	フィールドがガイドの実行に必要な場合は、 <b>【必須】</b> を選択します。

組み込みガイドでの入力フィールドの使用については、[Embedded Guide step \(ページ 128\)](#) を参照してください。

フィールドの詳細については、「[データ型とフィールドプロパティの概要](#)」(ページ 10) を参照してください。

## 出力フィールドのプロパティ

ガイドに対して使用できる出力フィールドは、プロセスの適用先のオブジェクトに含まれるフィールドです。これは、**【開始】** タブの **【適用対象】** プロパティで指定します。

出力フィールドの値は、ガイドの実行時に設定され、後続のステップ内で使用できるようになります。出力フィールドは、組み込みガイド内で使用する目的でのみ定義します。これらの出力フィールドは、値が組み込みガイドから返されるまではリストに表示されません。

次の表に、ガイドで使用する出力フィールドカテゴリの名前とタイプを示します。

プロパティ	説明
名前	出力フィールドの名前です。
タイプ	出力フィールドのタイプです。例えば、 <b>【日付】</b> 、 <b>【日付/時刻】</b> 、 <b>【時間】</b> 、 <b>【整数】</b> 、または <b>【テキスト】</b> を選択します。 単純型、カスタムタイプ、または定義された接続タイプの出力フィールドを追加することもできます。入力フィールドを追加するには、 <b>【タイプをさらに表示】</b> を選択し、 <b>【タイプの編集】</b> ダイアログボックスで、 <b>【カテゴリ】</b> に <b>【単純型】</b> 、 <b>【カスタムタイプ】</b> 、または <b>【定義された接続タイプ】</b> を選択します。

プロパティ	説明
説明	出力フィールドの説明です。
初期値	出力フィールドの初期値です。

組み込みガイドでの出力フィールドの使用については、[Embedded Guide step \(ページ 128\)](#)を参照してください。

フィールドの詳細については、「[データ型とフィールドプロパティの概要](#)」(ページ 10)を参照してください。

## 一時フィールドのプロパティ

プロセスに対して使用できる一時フィールドは、プロセスの適用先のオブジェクトに含まれるフィールドです。一時フィールドの値は、現在のガイドのすべてのステップ内で使用されます。ただし、一時フィールドの値は、組み込みガイド内で使用できるようにはなりません。

次の表に、ガイドで使用する一時フィールドカテゴリの名前とタイプを示します。

プロパティ	説明
名前	一時フィールドの名前です。
タイプ	一時フィールドのタイプです。例えば、[日付]、[日付/時刻]、[時間]、[整数]、または[テキスト]を選択します。 単純型、カスタムタイプ、または定義された接続タイプの一時フィールドを追加することもできます。入力フィールドを追加するには、[タイプをさらに表示]を選択し、[タイプの編集]ダイアログボックスで、[カテゴリ]に[単純型]、[カスタムタイプ]、または[定義された接続タイプ]を選択します。
説明	一時フィールドの説明です。
初期値	一時フィールドの初期値です。

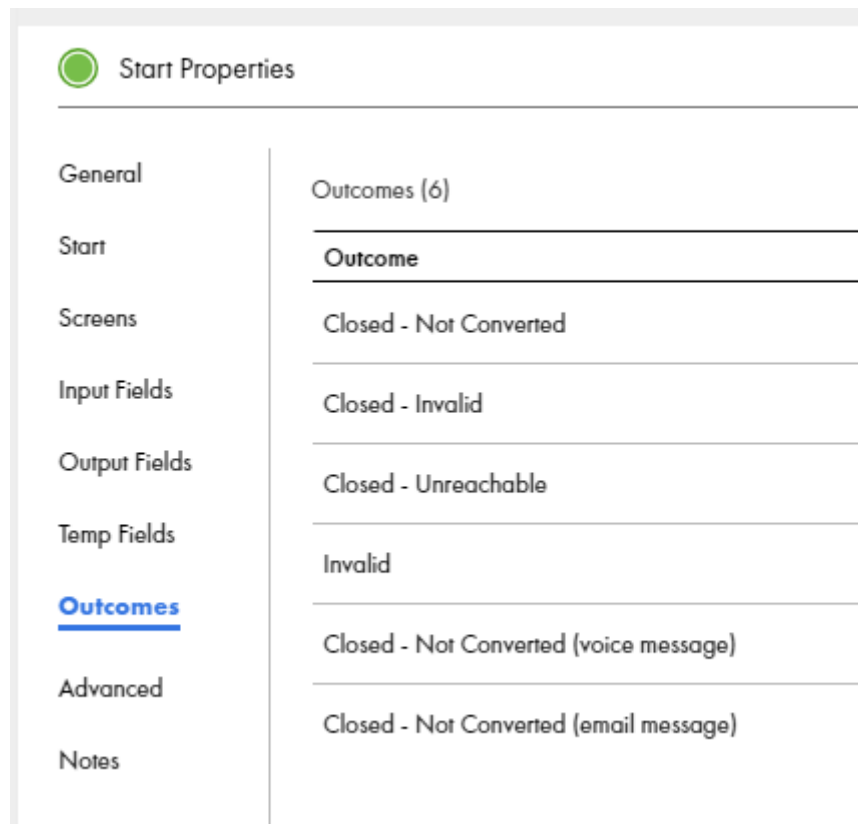
フィールドの詳細については、「[データ型とフィールドプロパティの概要](#)」(ページ 10)を参照してください。

## 結果

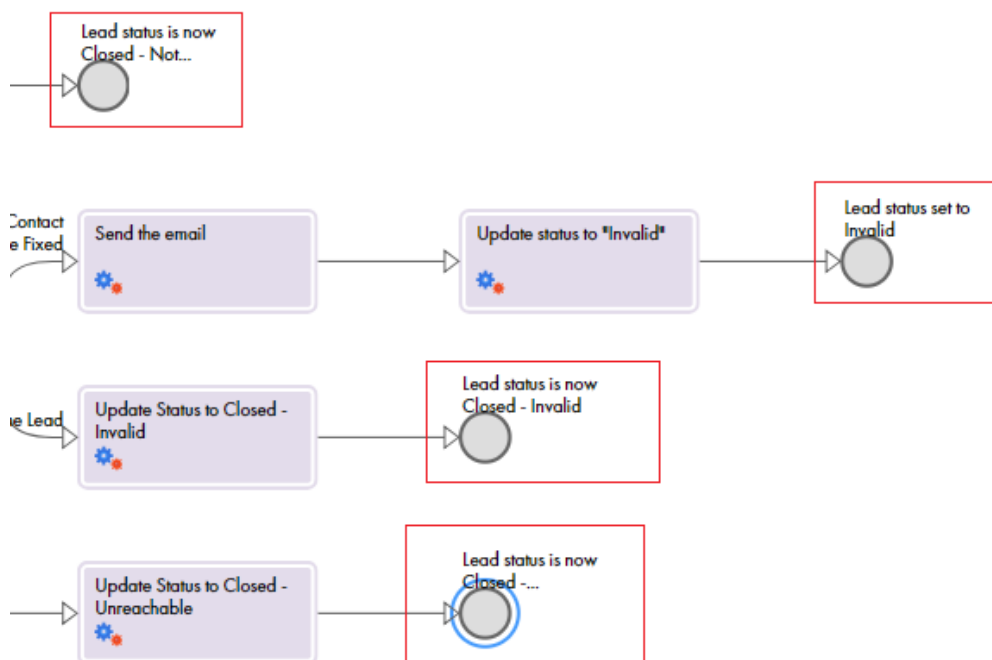
ユーザーがガイドの最後のステップを完了したときに、その終了方法をメモする必要がある場合があります。このような終了は「結果」と呼ばれます。

可能性がある結果を定義するには、[開始] > [結果]を使用します。

次の図では、ガイドに6つの結果が定義されています。



次の画像は、4つの結果があるガイドの一部を含むガイドのキャンバスビューを示しています。



編集集中も、使用可能な結果のリストで [End/Milestone Step \(ページ 131\)](#) **[新規結果を追加]** を選択することで、結果を追加できます。



## 詳細プロパティ

次のプロパティを設定するときに、[詳細] タブを使用します。

### 再開を許可

ユーザーがこのガイドを最初のステップから再開できるかどうかを決定します。このオプションを有効にすると、ガイドの再開時に、前に実行したステップを取り消します。

### 実行のロギング

選択した場合、アプリケーションの統合は、ガイドのステップと実行に関する詳細をログに記録します。そのため、この詳細はレポートに使用できます。

### ロールバック画面

選択した場合、ユーザーが [1 ステップ前に戻る] をクリックするか、[履歴] パネルでステップをクリックしたときに以前の入力値がリセットされます。

選択しない場合、以前の入力値はリセットされません。

組み込みガイドに対してこのオプションを選択した場合、親ガイドでもこのオプションを有効にしている場合を除き、リセットは組み込みガイドの範囲にのみ適用されます。

### 常に新規を開始

選択した場合、ガイドを開くたびに、そのガイドの新しいインスタンスが作成されます。

### フィードバック電子メールのリスト

ユーザーがフィードバックを送信するようにする場合、フィードバックを受信するための1つまたは複数の電子メールアドレスをカンマで区切って追加します。そうすると、ユーザーはガイドのフィードバックオプションをクリックして、次のダイアログを表示できます。

このダイアログには、ガイドの名前と、実行されるステップが表示されます。

### [次の場合のみガイドを表示します] (Salesforce のみ)

このガイドがユーザーの Salesforce ページの [ガイド] 領域に表示されるタイミングを決定する条件を指定できます。

次の図は、[詳細] タブを示します。

Start Properties

General

Start

Screens

Input Fields

Output Fields

Temp Fields

Outcomes

**Advanced**

Notes

Allow Restart

Execution Logging

Rollback Screens

Always Start New

Feedback Email List:

Show Guide Only When:  ▼

## 次の場合のみガイドを表示します。

**注:** この関数は、Salesforce 内で実行されるガイドにのみ適用されます。

ガイドのプロパティに条件を入力することによって、ユーザーの Salesforce ページの [ガイド] 領域内に表示されるガイドを決定できます。条件をガイドプロパティの [詳細] タブの [次の場合のみガイドを表示します] フィールドに入力します。

条件は Boolean 式として入力する必要があります。Salesforce のマニュアルに記載されている関数と演算子を使用できます。以下の例を参照してください。

**[フィールドの挿入]** リストを使用して、オブジェクトやフィールドを式に挿入することを選択します。

この式は、必要なだけ複合させることができます。例えば、Guide Designer に、アカウントの所有者が 3 人のうちの 1 人である場合のみ、ガイドを表示するように指示する式を作成できます。

```
CONTAINS($Account.Owner.Name, "John Smith")  
|| CONTAINS($Account.Owner.Name, "Bob Smith")  
|| CONTAINS($Account.Owner.Name, "Jim Smith")
```

**注:** フィールド名の入力時に、Salesforce のフィールド構文（中括弧内のフィールド名を「!」プレフィックスで囲む）を使用しないでください。式は、Apex 式として Salesforce に直接渡されます。無効な Apex 式を渡すと、Salesforce はエラーメッセージを返します。

### 例 1

フィールドがリストの場合は、等号を使用して、リストから 1 つの項目を選択できます。以下に例を示します。

```
Account.Type = 'Analyst'
```

### 例 2

複数の条件を定義するには、「&&」（論理 AND）と「||」（論理 OR）を式に使用します。以下に例を示します。

```
(Account.Type = 'Analyst' && Account.testfield__c= true) || (Account.Phone = '1234')
```

この例では、この式は次のことを意味します。

1. アカウントのタイプは Analyst である必要があり、かつ（AND）testfield は真である必要があり、または（OR）
2. アカウントの電話番号は 1234 である必要があり、

この場合、括弧は最初の 2 つのテストをグループ化するため、これらは 1 つの単位として評価されます。

次の例では、括弧によって、この式は異なって評価されます。

```
(Account.Type = 'Analyst') && (Account.testfield__c= true || Account.Phone = '1234')
```

ここで、同じ条件が次のことを意味します。

1. アカウントのタイプは Analyst である必要があり、かつ（AND）
2. testfield が真である、または（OR）アカウントの電話番号が 1234 であるかのいずれかである必要があり、

## 付記

**[付記]** タブを使用して、自分または他の開発者が必要とする可能性がある情報を追加します。入力した付記は、設計時にのみ表示されます。ガイドまたはプロセスのユーザーやコンシューマに付記は表示されません。例えば、プロセスやガイドをデプロイする前に、実行する必要がある事柄についての通知を追加できます。

# ガイドステップの追加

ステップを追加して、サービスやデータにアクセスしたり、複数の関連アクティビティを実行したりします。

ステップを追加するときに、そのステップのプロパティも定義します。

次のいずれかのタスクを実行して、ガイドにステップを追加します。

- 左側のパレットからキャンバスにステップをドラッグします。
- キャンバスをダブルクリックし、**[ステップのタイプ:]** リストからステップを選択します。

## 割り当てステップ

フィールドに値を設定するには、割り当てステップを使用します。**[割り当て]** ステップを作成するには、アクセスをクリックし、**[デザイン]** キャンバスから **[割り当て]** を選択します。次に、**[割り当て]** のプロパティパネルで **[追加]** アイコンをクリックして、**[名前]** と **[割り当て]** の値を割り当てます。

次の図は、**[割り当て]** ステップを示します。

Field	Assigned Using	From
input1	Formula	math:sqrt(100)
temp1	Field	input1
input2	Content	100
temp2	Specific date	06/15/2021 12:00 PM

フィールドを追加するには、**[追加]** アイコンをクリックし、各フィールドに次の情報を追加します。

### 名前

これは、完全修飾のフィールド名です。名前には、英数字、アンダースコア(\_)、スペース、および Unicode 文字のみを含めることができます。

フィールド名は、**[開始]** > **[フィールド]** で定義したフィールドの一覧から選択できます。

### 割り当て

これは、フィールドの値を取得するソースです。表示されるフィールドは、**[開始]** > **[プロパティ]** > **[入力フィールド]** または **[一時フィールド]** で定義したデータ型によって異なります。

例えば、入力フィールドのデータ型を **[日付]** または **[時間]** として定義すると、割り当てステップのそのフィールドに次のオプションが表示されます。

- 特定の日付
- 本日からの日数
- 前/後の日数
- 画面
- フィールド
- 計算式

データ型を整数またはテキストとして定義した場合、フィールドの値を指定するための次のオプションが表示されます。

- コンテンツ
- 画面
- フィールド
- 計算式

データ型の詳細については、「[\[フィールドのプロパティ\] ダイアログの使用](#)」(ページ 20)を参照してください。

## 作成ステップ

作成ステップを使用して、新しいオブジェクトインスタンスを追加します。例えば、アカウントオブジェクトを操作する場合、作成ステップを使用して、新しいアカウントを追加できます。

作成ステップ名には、英数字、アンダースコア (\_)、スペース、および Unicode 文字のみを含めることができます。

次の図は、作成ステップを示します。

Name	Required	Type
Account Name	<input checked="" type="checkbox"/>	Text
Account Type	<input type="checkbox"/>	Picklist
Record Type ID	<input type="checkbox"/>	Object ID (
Parent Account ID	<input type="checkbox"/>	Object ID (

新しいオブジェクトを作成するには、以下のタスクを実行します。

1. **【入力フィールド】** タブを選択して、オブジェクトに追加するフィールドを選択します。必須フィールドとオプションフィールドがあります。
2. ソースを **【コンテンツ】**、**【フィールド】**、**【計算式】**、または **【画面】** のいずれかの値に設定します。

**【画面の編集】** オプションを使用して、フィールドとラベルを配置するか、必要に応じてユーザーに入力を要求します。

### 参照フィールドの使用

参照フィールドについて、次の情報に注意します。

- 作成するオブジェクトに対する参照フィールドを **【入力フィールド】** タブに含めない場合、Guide Designer は、現在のオブジェクトの値を実行時に使用します。
- **【入力】** タブで参照フィールドが明示的に設定されない場合、**【適用対象】** オブジェクトの参照フィールドが設定されます。

- 参照フィールドのコンテンツが空に設定された場合、参照フィールドは作成文には含まれませんこれにより、明示的にフィールドが空に設定された場合、または NULL 値がデータの制約に違反する可能性がある場合に問題が発生することが防止されます。

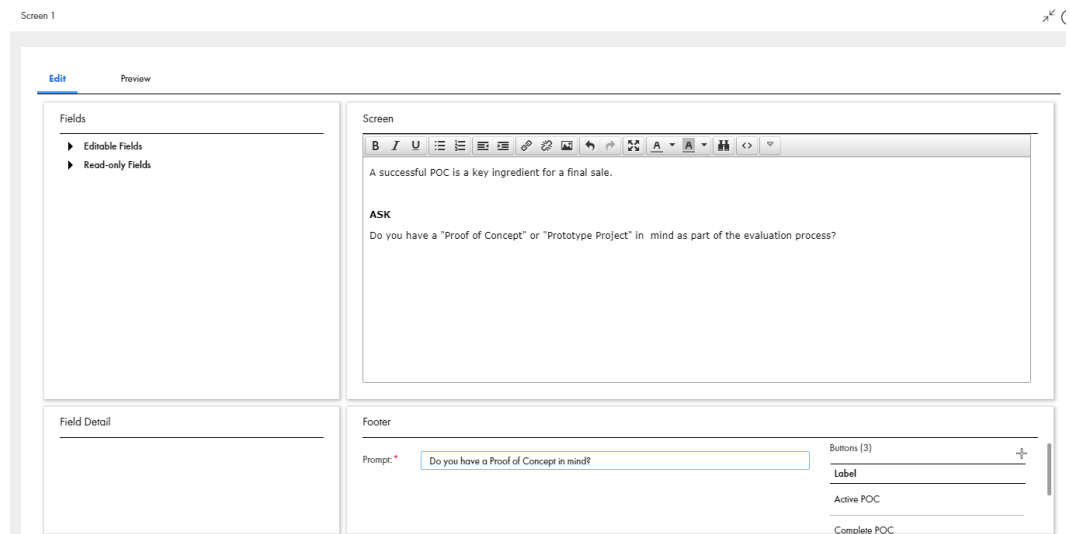
## 画面ステップ

画面ステップを使用して、次のタスクを実行します。

- 何が表示されているか、何を入力するか、などの指示を表示します。
- オブジェクトのデータをユーザー入力に基づいて更新します。

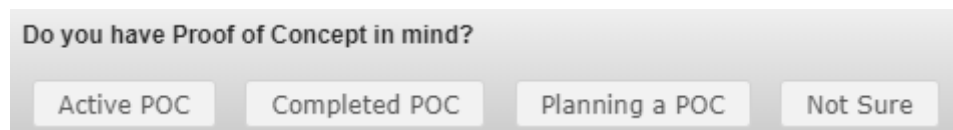
画面ステップ名には、英数字、アンダースコア ( \_ )、スペース、および Unicode 文字のみを含めることができます。

次の図は、営業担当者が見込み客に伝える内容が含まれる画面ステップを示します。



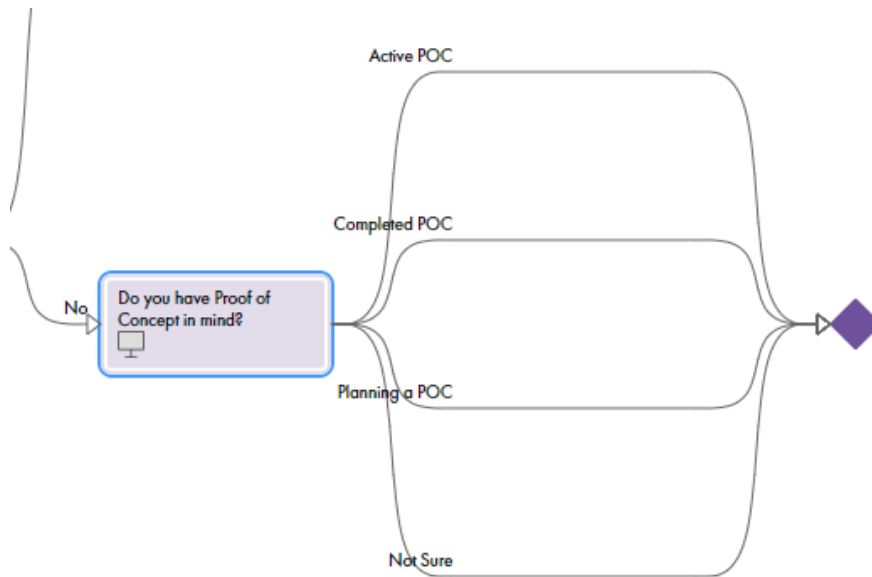
営業担当者がガイドを実行すると、確認が必要な質問と、それぞれに想定される回答のボタンが表示されます。

次の図は、シミュレートした画面ステップの一部を示します。

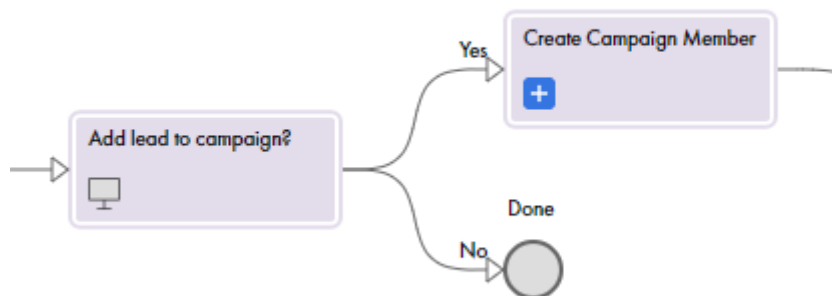


異なる選択で画面ステップを追加すると、Guide Designer は回答ごとに分岐を作成します。

次の図は、想定される 4 つの回答が表示された画面ステップを示します。



画面ステップで、分岐ごとにステップを追加できます。  
 次の図は、2つの回答が表示された画面ステップを示します。



画面ステップを編集するときに、次のフィールドタイプを挿入できます。

- 編集可能フィールド: ユーザーが編集できるフィールド。これらのフィールドに値を入力しなくても、次のステップに進むことができます。
- 読み取り専用フィールド: エンドユーザーが編集できないフィールド。

画面ステップの翻訳バージョンを作成するには、[Guide Translations \(ページ 113\)](#)を参照してください。

## 決定ステップ

以下の表に、決定ステップのプロパティを示します。

プロパティ	説明
名前	決定ステップの名前。名前には、英数字、アンダースコア(_)、スペース、および Unicode 文字のみを含めることができます。
決定	<p>ガイドは、ここで定義したフィールドとパスに基づいて決定を行います。</p> <p><b>【スタート】</b> &gt; <b>【フィールド】</b> で定義したフィールドの一覧からフィールド名を選択します。決定ステップで決定を下すために必要な条件と値を入力します。</p> <p>使用できる条件は、選択したフィールドによって異なります。</p> <p>例えば、<b>【簡易】</b> &gt; <b>【テキスト】</b> を選択した場合、次の条件が使用できます。</p> <ul style="list-style-type: none"><li>- 次の値に等しい</li><li>- 次で開始する</li><li>- 次で終わる</li><li>- 次のいずれかで開始する</li><li>- 次を含む</li></ul> <p>選択した条件に対してテキスト値を入力できます。</p> <p>決定ステップには複数の条件を追加できます。各条件は、潜在的なデータパスです。</p> <p>追加したパスごとに、対応する分岐が UI に表示されます。分岐をドラッグして、分岐が UI に表示される順序を並べ替えます。</p> <p>ほとんどの決定ステップには、別のパスがあります。テストの条件を満たしているデータがない場合、このパスが実行を処理します。</p>

### パスの評価

ガイドは、指定した基準に基づいて条件を評価します。パスは必ず、交差しない条件を指定して作成します。

例えば、次のパスで決定ステップを作成します。

- パス 1: フィールドは 100 以下とする。
- パス 2: フィールドは 75 以下とする。
- パス 3: フィールドは 25 以下とする。
- パス 4: それ以外。

決定ステップが作成された整数フィールドの値が 25 の場合、決定ステップはパス 1 を取ります。これは 25 が 100 未満で、パス 1 が最初のオプションであるためです。

データ決定ステップが「25 未満のフィールド以下」のパスに従っていることを確認するには、次の条件でパスを再作成します。

- パス 1: 0 から 25 の間の整数。
- パス 2: 26 から 75 の間の整数。
- パス 3: 76 から 100 の間の整数。
- パス 4: それ以外。

**重要:** ガイドは、トップダウン方式で条件を評価します。それ以外の分岐が最後のパスであることを確認します。

決定ステップは、別の決定ステップに連結できます。例えば、年間所得が 10 万ドルを超えると、分岐を実行できるなどと指定できます。同じパスの次の決定テストで、都市がボストンであるか、それ以外であるかをテストできます。この方法を使用するときには、最初の条件での真の分岐に基づいて 2 番目の条件をテストする

ため、ブール値 AND ロジックを使用します。この例では、決定ステップを使用して、「年間収入が 10 万ドルを超え、都市はボストンである」という AND 条件を設定します。

同様に、ブール型 OR ロジックをサポートするために、任意の分岐に 2 番目の条件のテストを追加できます。

## ジャンプステップ

ガイドを作成するときに、一連の選択でいくつかのアクティビティが必要になり、分岐が発生することがあります。これらのアクティビティの完了後、ステップのフローは別の分岐のアクションに再結合されます。

例えば、見込み客との対話を営業担当者が進める際のガイドで、営業担当者が製品や課題の目標について話すときに多くの異なる分岐に分かれる場合があります。ただし、これらの分岐の一部のあるポイントで、営業担当者はミーティングを設定する必要があります。スケジューリングステップと、それに続くステップが同じになる場合があります。スケジューリングが必要なランチごとにスケジューリングステップを追加する代わりに、スケジューリングステップにジャンプすることができます。

ジャンプステップを作成する際は、次のプロパティを設定します。

プロパティ	説明
送信先	ジャンプのターゲット。使用可能なステップの一覧から選択します。 複数のステップが同じターゲットステップにジャンプできます。ターゲットとして特定のステップがあるジャンプステップの数を確認するには、ターゲットステップの横にある矢印の上にカーソルを置きます。 ジャンプステップでは、並列パスステップと組み合わせて使用する場合、次の制限があります。 <ul style="list-style-type: none"><li>- 並列パスステップの外部から、並列パスステップのいずれかの分岐にジャンプすることはできません。</li><li>- 並列パスステップ内から並列パスステップの外側にジャンプすることはできません。</li></ul> 並列パスステップの 1 つの分岐から別の分岐にジャンプすることはできます。

## プロセス呼び出しステップ

プロセス呼び出しステップを使用して、ガイド内からプロセスを呼び出します。

プロセス呼び出しステップ名には、英数字、アンダースコア ( \_ )、スペース、および Unicode 文字のみを含めることができます。

ガイドはそれ自体にすべてが揃っている必要はありません。ほとんどのガイドは、次のように、外部で作成された一連のアクションを組み込みます。

- プロセス呼び出し。プロセス内で定義されたアクションは、現在のガイド内に配置されます。
- サービス呼び出し。サービスで定義されたアクションは、現在のガイド内に配置されます。
- 組み込みガイド。独立した別のガイドが、現在のガイド内に配置されます。

プロセス呼び出しステップを使用する方法は、組み込みガイドステップを使用する場合と同じです。詳細については、[Embedded Guide step \(ページ 128\)](#)を参照してください。

## 組み込みガイドステップ

組み込みガイドステップを使用して、ビジネスプロセスのオーケストレーションと管理をより効率的に行います。他のガイドが組み込まれたガイドをデザインします。これは、他のプロセスから呼び出されるサブプロセスを作成する場合と似ています。



組み込みガイドの使用には、次の利点があります。

- **繰り返しの回避:** 頻繁に使用されるパターンを個別のガイドに配置した場合、一連のステップを分離して1つのガイドに配置し、それを他のガイドに組み込んで再利用できるようにすることで、時間を節約してその他の効率性を追求できます。組み込みガイドを変更した場合に、それを使用するすべてのガイドでも、その変更を簡単に利用できます。
- **デザインの明確化:** 組み込みを利用することで、デザインがよりシンプルに見えるようになります。複数のステップを削除してそれらを組み込みガイドに配置することで、オブジェクトを組み込んだ結果、複数のステップすべてではなく1つのステップのみが表示されるようになります。これにより、キャンバス上の図を簡略化できます。また、問題が発生してステップの実行が妨げられる場合に、組み込みを行うことで、問題をより簡単に特定できます。
- **論理の明確化:** 他のオブジェクトで実行する必要があるステップがたくさんある場合、さまざまなオブジェクト（商談やリードなど）に適用される、作業の論理的な固まりを作成することができます。これは、1つのガイドで複数のオブジェクトタイプを扱う複雑さを低減します。

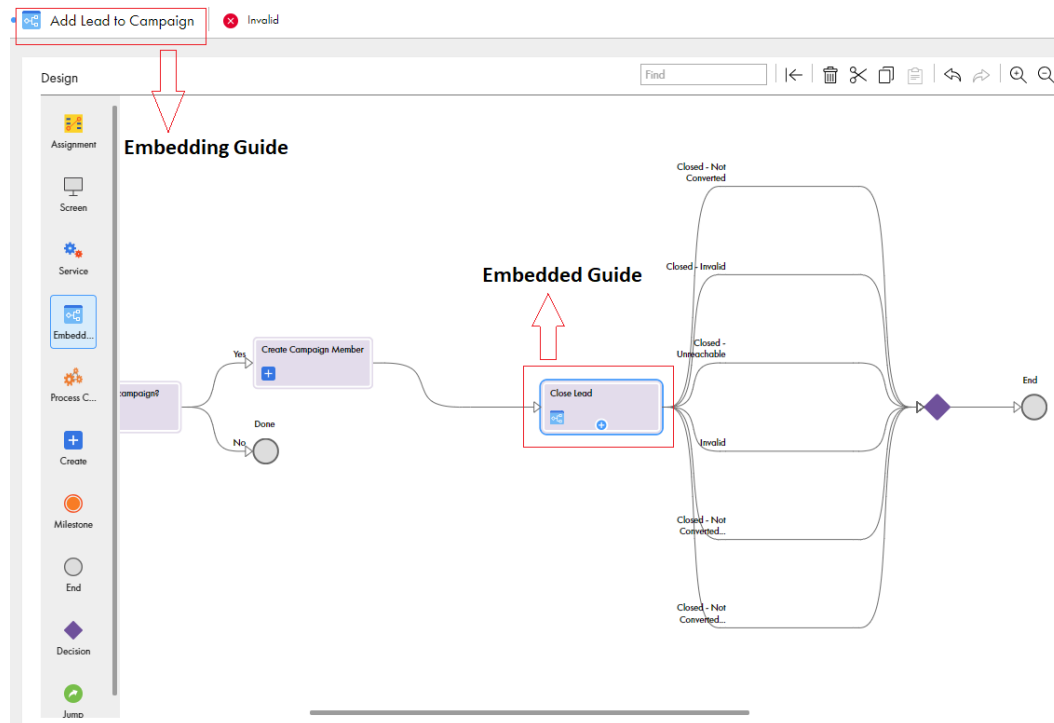
ガイドプロパティで定義する入力フィールドと出力フィールドを使用して、ガイドから受け取ったデータと、ガイドに送り返すデータを処理できます。

例として、クライアント情報の入力に関連するステップのセットがあるとします。ユーザーがクライアントと関わる方法が単一でなく、多くのユーザーがクライアント情報を入力する必要がある場合、これらのクライアント情報のステップを専用のガイドにグループ化できます。このようにすることで、ガイドでクライアント情報が必要になったときはいつでも、ガイドを「組み込む」だけで済むようになります。また、このことは、組み込み側のすべてのガイドが同じステップを使用することも意味します。

組み込みガイドステップを作成するとき、次のプロパティを入力します。

プロパティ	説明
全般	組み込みガイドステップの名前です。名前には、英数字、アンダースコア(_)、スペース、およびUnicode文字のみを含めることができます。
ガイド	現在のガイドに組み込むガイドを選択します。 現在のガイドは、組み込み側のガイドと呼ばれます。 組み込み側のガイドが組み込むガイドは、組み込みガイドと呼ばれます。
反復ルール	組み込みガイドを、単一のオブジェクト上で実行するか、すべてのオブジェクトで実行するかを設定します。 <b>[単一オブジェクトで実行]</b> を選択した場合、組み込みオブジェクトは、入力として受け取る特定のオブジェクト上で実行されます。組み込みガイドは、実行が完了したときに、組み込み側のガイドに続くステップに制御を渡します。 <b>[次のタイミングまでリスト内の各オブジェクトで実行]</b> を選択した場合、ガイドの実行を制御するイベントも選択します。 ランタイムに処理する次のオブジェクトから選択することもできます。 - <b>[フィールド]</b> : 1つのオブジェクトまたはオブジェクトのリストが組み込みガイドに渡されます（選択した実行の種類に基づく）。 - <b>[クエリ]</b> : [単一オブジェクトで実行] でのみ使用できます。組み込みガイドに渡すオブジェクトを選択する WHERE 条件を定義できます。 - <b>[リスト]</b> : [次のタイミングまでリスト内の各オブジェクトで実行] の結果でのみ使用できます。組み込みガイドに渡すオブジェクトを選択する WHERE 条件を定義できます。
入力フィールド	選択したオブジェクトで使用できる入力フィールドを表示します。組み込みガイドがこれらのフィールドのいずれかを入力として受け取った場合、組み込みガイドの実行中に、このフィールドの値が更新されることがあります。その場合、この組み込みガイドの実行が完了してから、親ガイドの値が更新されます。

次の図は、組み込みガイドが含まれる組み込み側のガイドを示します。



## サービスステップ

サービスステップを使用して、外部サービス、システムサービス、またはプロセスに接続します。  
 サービスステップを追加するときは、サービスタイプ、接続、およびアクションを設定する必要があります。  
 次の表では、サービスステップのプロパティについて説明しています。

プロパティ	説明
全般	サービスステップの名前。名前には、英数字、アンダースコア(_)、スペース、および Unicode 文字のみを含めることができます。
サービスタイプ	プロセスに追加する接続、プロセス、またはシステムサービス。既存のタスクの一覧から選択します。 <small>注: ガイドに追加する既存の項目が必要です。サービスステップを使用する場合は、項目を作成できません。</small>
接続	サービスコネクタへの接続。
アクション	接続に含まれるアクション。

次の図は、Salesforce 接続が選択されたサービスステップを示します。

Submit For Approval Properties

General

Service Type: Connection

Connection: Default > Salesforce

Input Fields

Action: Submit For Approval

Description

Submits an object for approval based on the approval process associated with the object.

Input Fields

Name	Required	Type	Description
Object to approve	<input type="checkbox"/>	Object ID	The object to submit for approval, by default it is the object the guide is running on (applies to object). You can optionally specify an object other than the current object for example an Account, Contact, or custom object.
Approver	<input type="checkbox"/>	Object ID (Salesforce:User)	If the approval process requires manual selection of the approver then this should be the ID of the user for the request.
Comments	<input type="checkbox"/>	Text	Optional comments to add to the submission.

Output Fields

## 入力フィールド

**【入力フィールド】** セクションに、この種類のステップを使用するときに最も多く使用される入力フィールドの名前が表示されています。オブジェクトを作成するサービスステップの場合、表示される入力フィールドは、オブジェクトの作成時に最も頻繁に必要なフィールドになります。(必要でないオプションのフィールドは、削除できます。) 追加の (オプション) 入力フィールドをリストから選択できます。

削除アイコンを使用して、フィールドを削除します。これにより、サービスステップに渡さない入力フィールドが削除されます。一部のフィールドは必須で、削除できません。

## 終了/マイルストーンステップ

終了/マイルストーンステップは、ガイドの実行またはマイルストーンの終了をマークします。

ガイドに終了ステップを追加するときに、そのステップを **【マイルストーン】** タイプのステップにするか、**【ガイドの終了】** タイプのステップにするかを設定できます。マイルストーンの終了は、ガイドのアクティビティに含まれる 1 つのフェーズの終了を示します。

**注:** テーブルにテキストやフィールドを挿入すると、入力するテキストをより読みやすくなります。詳細については、「[テーブルのフィールドの挿入](#)」を参照してください。

### ガイドの終了ステップ

ガイドの終了ステップは、ガイドが実行を終了する場所をマークします。ガイド内のすべてのブランチは、[Jump Step \(ページ 128\)](#) でない限り、終了ステップで終了します。ガイドの終了ステップは、結果を指定しません。

次の図は、**【ガイドの終了】** ステップのサンプルです。

End Properties

General

Ending Type:

Milestone

End of Guide

Show Screen: [Edit Screen](#) [Preview Screen](#)

Outcome: Evaluation Sent

Preprocessing: Go to Other Object

Master Record > Owner > Group

**【ガイドの終了】** オプションを選択するときに、次のプロパティを設定します。

- **【終了タイプ】** : マイルストーン

- **【画面に表示】**：ユーザーにマイルストーンステップを表示するには、このオプションを選択します。選択しないと、マイルストーンステップはユーザーに表示されません。代わりに、完了ボタンが押されたときのアクションを示す領域に表示されたアクションが、このステップの実行後すぐに行われます。
- **【結果】**：開始ステップで定義した結果が含まれるリストです。新しい結果を追加することもできます。
- **【前処理】**：このリストからオプションを選択して、マイルストーンステップの完了後に実行するアクションを設定します。
  - **【現在のオブジェクトを更新】**：ガイドを起動したページが更新され、ガイドによって実行されたアクションがオブジェクトのページに表示されるようになります。
  - **【他のオブジェクトに移動】**：選択すると、右側に **【クリックしてフィールドを選択】** リストが表示されます。この中から項目を1つ選択します。多くは、現在のオブジェクトに関連付けられたIDです。これは、新しいオブジェクトを作成することを主な目的としているガイド（リードなど）に特に有用です。ガイドの実行が終了した後にユーザーがオブジェクトに移動するように、終了ステップで指定することができます。
  - **【URLに移動】**：選択すると、右側に **【クリックしてフィールドを選択】** リストが表示されます。タイプがURLのフィールドを選択します。

## マイルストーンステップ

マイルストーンは、ユーザーが作業を他の人に渡すときに特に便利です。（ガイドが渡されるときに、ガイドを渡す側のユーザーは、何らかの作業が行われる必要があることを次のユーザーに知らせる必要があります。これを、アプリケーションのワークフローに組み込むことができます。）そのユーザーがガイドを開始すると、マイルストーンに続くステップから開始されます。また、ユーザーが他のアクションを行うために、あるステップでガイドの実行を停止する場合にも、マイルストーンを使用します。

いくつかのケースでは、単純に、発生したイベントを記録するためにマイルストーンステップを使用することができ、ガイドは次のステップを自動的に継続します。

次の図は、**【マイルストーン】** ステップのサンプルです。

**Milestone Properties**

General

**Milestone**

Ending Type:

- Milestone
- End of Guide

Show Screen: [Edit Screen](#) [Preview Screen](#)

Allow user to go back:

Preprocessing

- Refresh Current Object
- Go to Other Object
- Go to URL
- Refresh Current Object and Continue
- Continue

ガイドがマイルストーンに到達した後、それ以降のアクティビティが実行される必要はありません。ただし、14日以内に追加のアクションが発生しない場合、ガイドは終了します。**【マイルストーン】** ガイドのないガイ

ドは、7日間で終了します。マイルストーンに到達した後でアクションが開始されると、ガイドが終了するか、7日以内に別のマイルストーンに到達しない場合、ガイドは終了します。

**[マイルストーン]** オプションを選択するときに、次のプロパティを設定します。

- **[終了タイプ]** : マイルストーン
- **[画面に表示]** : ユーザーにマイルストーンステップを表示するには、このオプションを選択します。選択しないと、マイルストーンステップはユーザーに表示されません。代わりに、完了ボタンが押されたときのアクションを示す領域に表示されたアクションが、このステップの実行後すぐに行われます。
- **[戻れることをユーザー ID に許可:]** : マイルストーンの前に実行されたステップにユーザーが戻れることを許可する場合は、このオプションを選択します。いずれの場合も、ガイド内で発生したすべてのステップに、ガイドの履歴が表示されます。ただし、ユーザーが戻れない場合、ユーザーは以前に実行されたステップの1つを選択することはできません。
- **[前処理]** : このリストからオプションを選択して、マイルストーンステップの完了後に実行するアクションを設定します。
  - **[現在のオブジェクトを更新]** : ガイドを起動したページが更新され、ガイドによって実行されたアクションがオブジェクトのページに表示されるようになります。
  - **[他のオブジェクトに移動]** : 選択すると、右側に **[クリックしてフィールドを選択]** リストが表示されます。この中から項目を1つ選択します。多くは、現在のオブジェクトに関連付けられたIDです。これは、新しいオブジェクトを作成することを主な目的としているガイド（リードなど）に特に有用です。ガイドの実行が終了した後にユーザーがオブジェクトに移動するように、終了ステップで指定することができます。
  - **[URL に移動]** : 選択すると、右側に **[クリックしてフィールドを選択]** リストが表示されます。タイプがURLのフィールドを選択します。
  - **[現在のオブジェクトを更新して続行]** : このガイドが実行されているオブジェクトが更新された後、後続のステップに実行が移動されます。
  - **[続行]** : マイルストーンに到達し、実行が次のステップに続くことの注記です。

## ガイドの外観の設定

ガイドの外観をカスタマイズできます。組み込みのテーマから選択して、ユーザーがデスクトップのブラウザでガイドを実行するときにガイドがどのように表示されるかを決めることができます。ガイドの実行時に表示される手順をカスタマイズすることもできます。

### ガイドの外観

**[ガイドの設定]** ページを使用して、組織レベルで、ガイドのテーマを指定し、ガイドのイメージを管理します。

**[ガイドの設定]** ページ上のタブを使用して、ガイド設定を構成します。

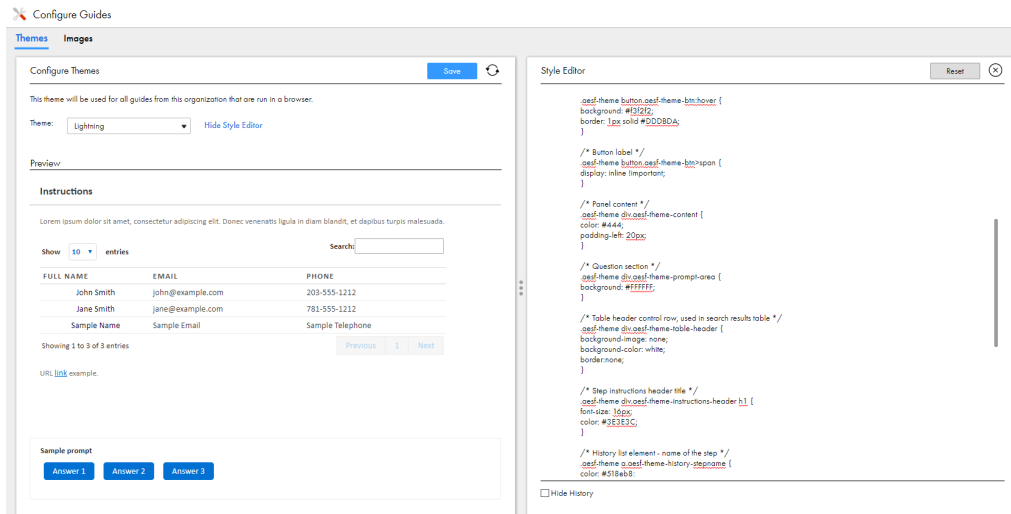
#### テーマ

ガイドを設定するためのテーマを組み込みテーマのリストから選択します。次のいずれかを選択できます。

- ブルーレイ
- ライトニング
- レドモンド

- ベイダー
- デフォルト

次の図は、[ガイドの設定] ページの [テーマ] タブで [ライトニング] テーマが選択されている状態を示しています。



[スタイルエディタの表示] をクリックして、テーマに含まれるガイドステップ、パラグラフ、または文字の CSS を表示および編集します。エレメントの種類で通常使用出来る任意の CSS プロパティを使用出来ます。

例えば、ステップのヘッダーテキストの色を青に変えるには、次の CSS を設定します。

```

aesp-theme div.aesp-theme-instructions-header h1 {
  color:blue;
}

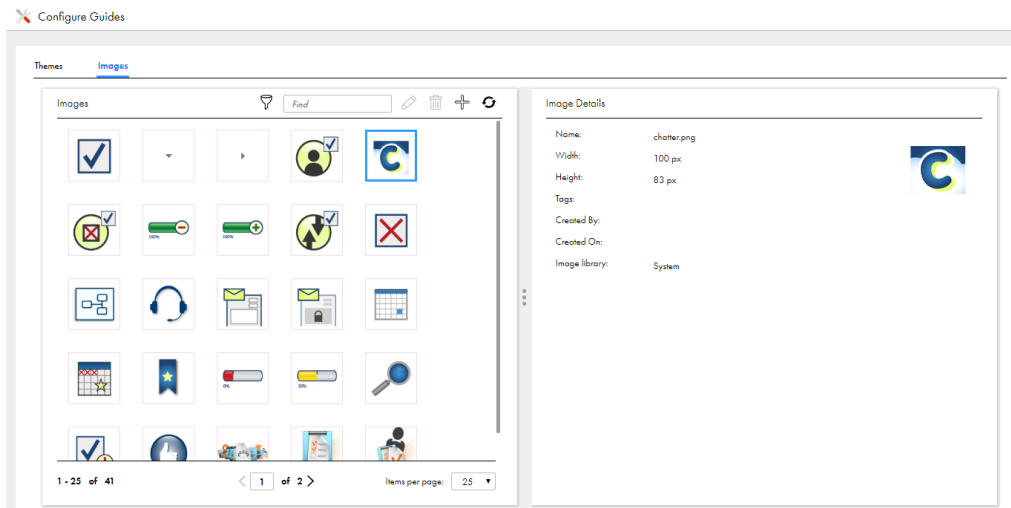
```

テーマを選択し、設定して保存すると、そのテーマは組織内のすべてのガイドに適用されます。

## 画像

[イメージ] タブを使用して、ガイドで使用するアイコンを表示および編集します。






次の図は、[ガイドの設定] ページで [イメージ] タブが選択されている状態を示します。





[イメージ] タブ最上部の [イメージの追加] をクリックすると、[イメージの追加] ダイアログボックスが表示されます。システムの.gif、.jpg、.jpeg、または.png イメージを、アプリケーションの統合のイメージライブラリにアップロード出来ます。

## ガイドの画面エディタ

画面エディタを使用して、ガイドに表示される画面に手順やフィールドを挿入し編集できます。画面エディタでは、次のツールを使用できます。

ボタン	説明
 (太字)	選択したテキストを <b>太字</b> に設定するか、入力しようとしているテキストが太字になることを示します。太字のテキストが選択されているときにこのボタンを押すと、太字が解除されます。(キーボード: Ctrl+B)
 (斜体)	選択したテキストを <i>斜体</i> に設定するか、入力しようとしているテキストが斜体になることを示します。斜体のテキストが選択されているときにこのボタンを押すと、斜体が解除されます。(キーボード: Ctrl+I)
 (下線)	選択したテキストに <u>下線</u> を付けるか、入力しようとしているテキストに下線が付くことを示します。下線の付いたテキストが選択されているときにこのボタンを押すと、下線が解除されます。(キーボード: Ctrl+U)
 (箇条書きリスト)	現在のパラグラフに箇条書きが設定されていない場合は、箇条書きにしてパラグラフのインデントを変更します。パラグラフが箇条書きの場合、このコマンドは、箇条書きを解除し、インデントのレベルを1つ減らします。 パラグラフ内または末尾で Enter キーを押すと、次のパラグラフも箇条書きになります。箇条書きリストを終了するには、改行を2回入力するか、このアイコンを再度選択します。
 (番号付きリスト)	現在のパラグラフに番号が設定されていない場合は、番号を追加してパラグラフのインデントを変更します。パラグラフに番号が付いている場合、このコマンドは、番号を解除し、インデントのレベルを1つ減らします。 先行するパラグラフが番号付きリストの一部である場合、このコマンドを選択すると、現在のパラグラフがそのリストに追加されます。 パラグラフ内または末尾で Enter キーを押すと、次のパラグラフにも数字が付きます。番号付きリストを終了するには、改行を2回入力するか、このアイコンを再度選択します。
 (インデントを減らす)	現在のパラグラフまたは選択したパラグラフにインデントが設定されている場合に、インデントのレベルを1つ減らします。
 (インデントを増やす)	現在のパラグラフまたは選択したパラグラフに、インデントのレベルを1つ追加します。

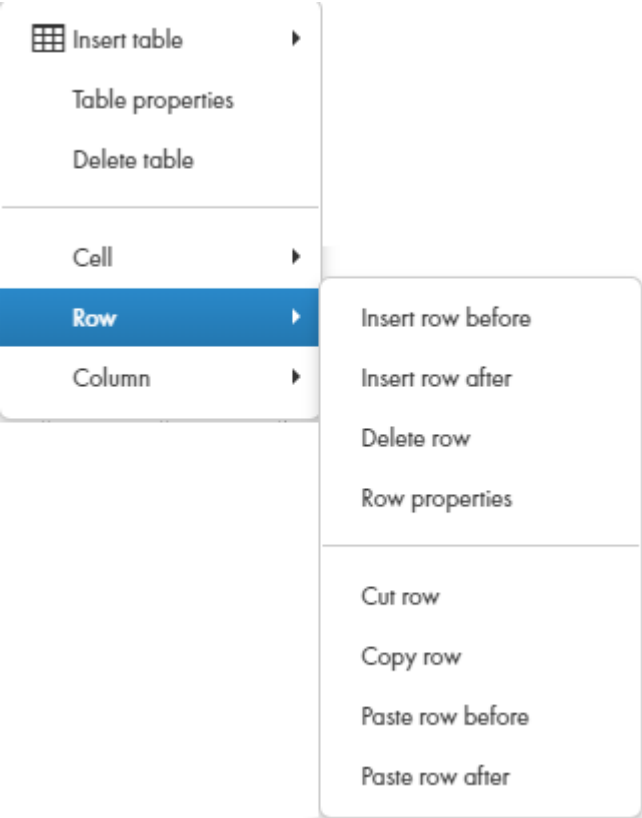






ボタン	説明
 (リンクの挿入/編集)	<p>選択したテキストをリンクに変更します。このボタンを選択すると、Guide Designer は次のダイアログを表示します。</p> <div data-bbox="464 394 1360 667" style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;"> <p style="text-align: right;">✕</p> <p>Insert Link</p> <p>Link URL: <input type="text"/></p> <p>Title: <input type="text"/></p> <p style="text-align: right;">Insert Link Cancel</p> </div> <p>このダイアログには、次のものが表示されます。</p> <ul style="list-style-type: none"> <li>- <b>[リンクの URL]</b> :ここに、Web アドレスを入力します。</li> <li>- <b>[タイトル]</b> :ユーザーがリンクにカーソルを重ねると表示されるテキストです。(このテキストは、一部のブラウザで表示されない場合があります。)</li> </ul> <p>カーソルがリンクの任意の場所にある間にこのボタンを押すと、リンクの情報を編集できます。</p> <p>テキストを選択するのを忘れると、リンクは表示されません。ただし、リンクが作成されます。これを修正するには、&lt;&gt;ボタンを押してリンクを特定します。HTML がわかる場合は、リンクを直接編集できます。わからない場合は、削除します。(空のリンクのテキストは、「&lt;a href=""&gt;&lt;a&gt;」のようになっています。)</p>
 (リンク解除)	<p>以前に作成されたリンクを解除します。カーソルは、リンク内の任意の場所に置くことができます。表示するテキストフィールド内のテキストは、このボタンを選択した後も残ります。</p>



ボタン	説明
 (画像の挿入/編集)	<p>画像が選択されていない場合、このボタンを押すと新しい画像を挿入できます。既存の画像やそのプロパティを変更するには、このボタンを押す前に画像を選択します。このボタンを押すと、Guide Designer は次のダイアログを表示します。</p>  <p>上部領域に表示される画像は、[デザインホーム] ページ上にある <b>【設定】</b> ボタンを使用して、以前に Guide Designer にインポートされたものです。また、現在の画面ステップで使用する画像をアップロードすることもできます。ここでアップロードした画像は、このガイド内の他のステップや、他のガイドでは使用できません。</p> <p>Web 上で使用可能な画像を、ステップに挿入することもできます。これは、Web サイトの画像を、画面エディタの挿入先に直接ドラッグして行います。</p> <p>このダイアログには、次のフィールドがあります。</p> <ul style="list-style-type: none"> <li>- <b>検索領域</b>: 左側のテキストボックスを使用して、[タイトル] フィールドまたは [説明] フィールドのいずれかに入力したテキストに基づいて画像を検索します。右側のテキストボックスを使用して、画像にラベルを付けるために使用されたタグを入力します。タグのテキストを入力すると、入力している文字と一致するタグが表示されます。</li> <li>- <b>[URL]</b>: URL を使用してアクセスできる場合は、ここに入力します。</li> <li>- <b>[タイトル]</b>: 画像のタイトルです。</li> <li>- <b>[説明]</b>: 画像を説明するテキストです。このダイアログに追加されたテキストを確認します。</li> <li>- <b>[幅 x 高さ]</b>: これらのフィールドを使用して、画像の幅と高さを変更します。ピクセル単位でサイズを入力するか、元のサイズに対する割合を入力できます。入力する値は、画像の元の寸法より小さくすることも大きくすることも可能です。幅のみをピクセル単位で入力した場合、サイズは比例して変更されます。例えば、前の例の表示幅を 235 に変更した場合、表示の高さはブラウザに応じて 93 または 94 になります。値を割合で入力した場合、その割合は高さにも適用されます。</li> <li>- <b>[元の W x H]</b>: 保存されたとおりの画像の幅と高さです。</li> <li>- <b>[float]</b>: この選択リスト内のオプションを使用して、挿入された場所との関連で画像の位置を示します。[なし] (1 列に表示)、[左] (画像は左にフロート)、[右] (画像は右にフロート) の選択肢があります。</li> <li>- <b>[参照]</b>: このボタンを押すと、コンピュータのファイルエクスプローラを使用して画像を選択できるダイアログが表示されます。選択してその内容を確認すると、ここに画像が表示されます。</li> </ul>
 (元に戻す)	<p>以前に実行したアクションが解除されます。(キーボード: Ctrl+Z)</p>
 (やり直し)	<p>以前に <b>【元に戻す】</b> を選択していた場合にこのボタンを押すと、変更が復元されます。(キーボード: Ctrl+Y)</p>

ボタン	説明
 (全画面)	ステップのサイズを大きくして、Guide Designer のタブの下の領域全体を占める大きさにします。すでに大きくしている場合にこのボタンを押すと、ステップが元のサイズに戻されます。 <b>注意:</b> エディタが全画面モードの場合は、ステップにフィールドを挿入できません。
 (テキストの色)	このボタンを押すと、テキストを表示するときに使用できる色のパレットが表示されます。テキストが選択されていない場合にこのボタンを押して色を選択すると、入力するテキストの色が変わります。このテキストからカーソルを離すと、色はカーソルがあるところの色になります。 このボタンを押して、テキストが選択された状態で色を選択すると、選択したテキストの色が変わります。
 (背景色)	このボタンを押すと、テキストの背後の色に使用できる色のパレットが表示されます。テキストが選択されていない場合にこのボタンを押すと、入力するテキストの背景色が変わります。このテキストからカーソルを離すと、背景色はカーソルがあるところの色になります。 このボタンを押して、テキストが選択された状態で色を選択すると、選択したテキストの背景色が変わります。
 (検索と置換)	このボタンを押すと、現在の画面内で特定するテキストを入力できるダイアログが、Guide Designer に表示されます。 [置換] フィールドにテキストを入力して置換ボタンを押すと、選択したテキストが置き換えられます。[すべて置換] を選択した場合、見つけられるすべてのテキストが置き換えられます。
 (ソースコード)	テキストが表示されている場合にこのボタンを押すと、Guide Designer はテキストのフォーマットに使用されている HTML を別のブラウザウィンドウで表示します。このブラウザウィンドウは、HTML ソースエディタです。これは切り替えボタンではありません。ソースエディタでの変更が完了したら、[OK] ボタンを押します。
 (他のツールを表示/非表示)	画面エディタのボタンに 2 つ目の行が追加されます。2 つ目の行が表示されている場合は、このボタンを押すと非表示になります。
 (フォントファミリー)	このボタンを押すと、テキストを表示するときに使用できるフォントの選択リストが表示されます。テキストが選択されていない場合にこのボタンを押してフォントファミリーを選択すると、入力するテキストのフォントが変わります。このテキストからカーソルを離すと、フォントはカーソルがあるところのフォントになります。 このボタンを押して、テキストが選択された状態でフォントを選択すると、選択したテキストのフォントが変わります。
 (フォントサイズ)	このボタンを押すと、テキストを表示するときに使用できるフォントサイズの選択リストが表示されます。テキストが選択されていない場合にこのボタンを押してサイズを選択すると、入力するテキストのサイズが変わります。このテキストからカーソルを離すと、フォントサイズはカーソルがあるところのサイズになります。 このボタンを押して、テキストが選択された状態でサイズを選択すると、選択したテキストのサイズが変わります。
 (テーブル)	このボタンを押すと、Guide Designer はテーブルコマンドの選択リストを表示します。[テーブルの挿入] を選択すると、ボックスのグリッドが表示されます。マウスを使用して、テーブル内に表示する行と列の数を選択します。 カーソルがテーブル内にある場合、この選択リストで他のコマンドを選択できます。

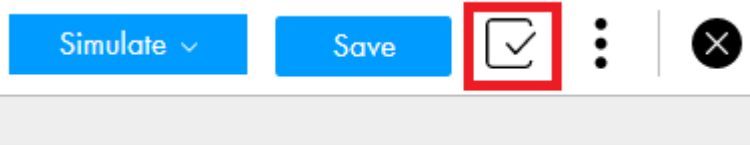
ボタン	説明
<p>テーブルプロパティ</p>	<p>選択リストから [テーブルプロパティ] を選択すると、2つのタブを持つダイアログボックスが表示されます。</p>  <p>このダイアログボックスを使用して、テーブルの高さと幅を表示します。(幅は、パーセント記号を使用して割合として設定することをお勧めします。) 高さはあまり設定しません。[セルの間隔] は、テーブルセル間のスペースを設定します。[セルの余白] は、セルのコンテンツとセルの端の間のスペースの広さです。枠線を設定する場合は、その幅を示す数を入力します。テーブルにキャプションを設定する場合は (通常、サイズが小さいステップのキャンバスには設定しません)、このボタンを選択します。テーブル上部に小さい領域が表示されます。最後に、[整列] 選択リストで [なし]、[左]、[中央]、または [右] を選択します。[なし] と [中央] が最もよく使用されます。</p> <p>[詳細] タブでは、テーブルのスタイル、枠線、および背景色を設定できます。</p> <p><b>[テーブルの削除]</b> コマンドを選択して、キャンバスからテーブルを削除します。</p>
<p>セルのプロパティ</p>	<p>[テーブル] 選択リストから [セル] を選択すると、2つ目の選択リストが表示されます。この選択リストを使用して、セルのプロパティ定義、2つのセルのマージ、マージされたセルの分割などを行えます。</p> <p>セルのプロパティを定義するときに、セルの幅や高さ (通常高さは選択しません)、タイプ (セルまたはヘッダー)、変更の範囲 (なし、行、列、行グループ、列グループ)、セルコンテンツの整列 (水平と垂直) を指定できます。</p>

ボタン	説明
行のプロパティ	<p>[テーブル] 選択リストから [行] を選択すると、2 つ目の選択リストが表示されます。</p>  <p>行のプロパティは、セルのプロパティと似ています。この選択リストのその他のコマンドは、コマンド名のとおり動作します。</p>
 (書式のクリア)	<p>テキストの表示方法が変更された場合にこのボタンを押すと、画面エディタは変更を解除します。</p>
 (左寄せ)	<p>現在選択しているパラグラフまたはカーソルが置かれているパラグラフを、左マージンに揃えます。</p>
 (中央揃え)	<p>現在選択しているパラグラフまたはカーソルが置かれているパラグラフを、左マージンと右マージンの中央に配置します。</p>
 (右寄せ)	<p>現在選択しているパラグラフまたはカーソルが置かれているパラグラフを、右マージンに揃えます。</p>
 (行揃え)	<p>現在選択しているパラグラフまたはカーソルが置かれているパラグラフの単語から行の間にスペースを追加して、テキスト行が揃うようにします。つまり、すべてのテキストが左マージンと右マージンに揃えられます。</p>
 (引用)	<p>選択しているパラグラフまたはカーソルが置かれているパラグラフのインデントを変更します。パラグラフはすでに引用に含まれる場合は、元のマージンが復元されます。</p>

# ガイドの検証と保存

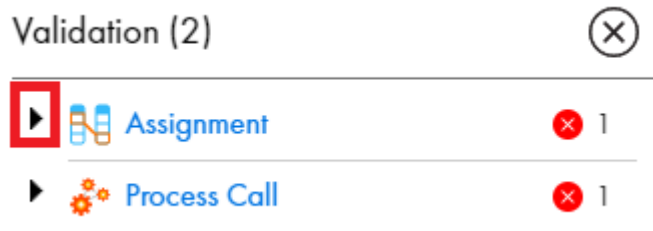
ガイドを作成する際に【検証】パネルを使用して、検証エラーの有無を調べたり、エラーの詳細を確認したりすることができます。

1. 次の図に示すように、ツールバーの【検証】アイコンをクリックします。



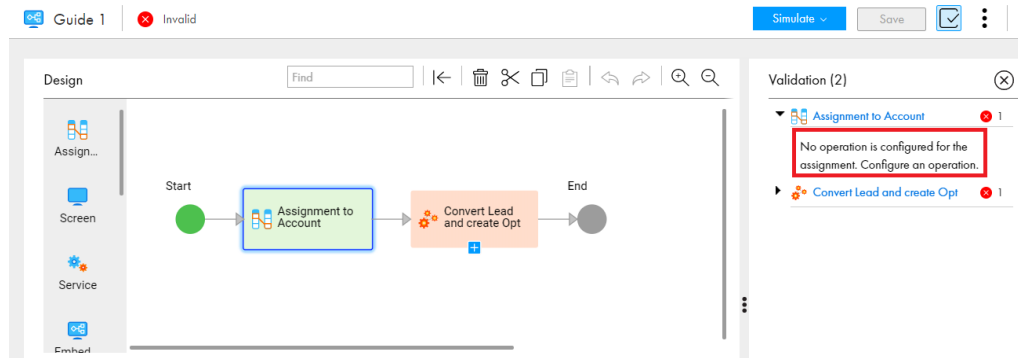
右ペインに【検証】パネルが表示されます。【検証】パネルでは、発生したタブに基づいてエラーが分類されています。

2. 次の図に示すように、タブ名に対応する矢印をクリックします。



タブで発生したエラーのリストが表示されます。

3. 次の図に示すように、エラー行をクリックします。



エラーを修正すると、【検証】パネルのエラーリストが更新されます。

4. ガイドを検証すると、ガイドを保存できるようになります。  
次の図に示すように、ツールバーの【保存】をクリックします。

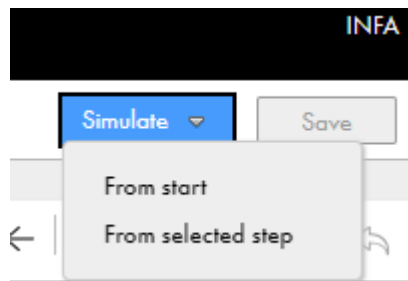


# ガイドのシミュレーション

Guide Designer の [シミュレート] オプションを使用して、ガイドを実行したときにガイドがどのように表示されるかを確認します。

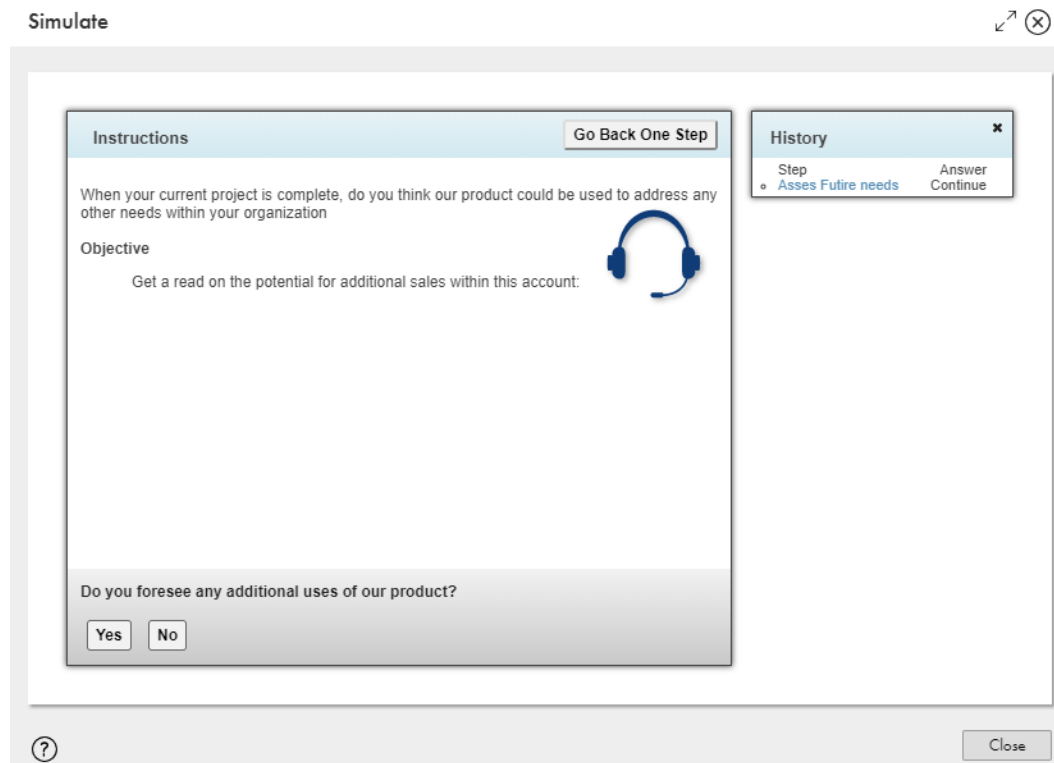
Guide Designer でのガイドのシミュレーションは、先頭からまたは任意のステップから実行できます。

次の画像は、[シミュレート] オプションを示しています。

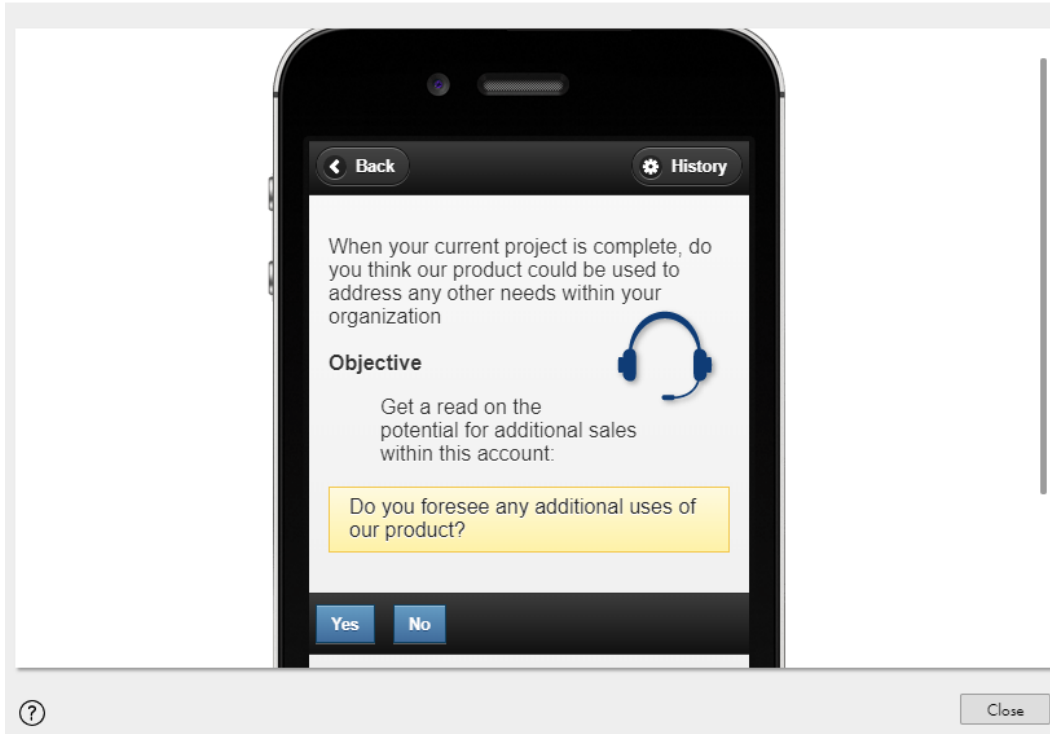


表示されるシミュレーションの種類は、[開始] > [画面] > [意図される表示:] で [デスクトップ/ターゲット] を設定したか、[スマートフォン] を設定したかによって異なります。

次の図は、表示が [デスクトップ/ターゲット] として設定されたガイドのシミュレーションを示します。



次の図は、表示が [スマートフォン] として設定されたガイドのシミュレーションを示します。



## 第 5 章

# プロセスオブジェクトのデザイン

ビジネスプロセスを設計するために、構造化データのセットを取得する必要があることがよくあります。例えば、人口統計的データの種類について個別の項目を作成する代わりに、名前、住所、電話番号のフィールドが含まれたグループを設けられます。これらのグループはプロセスオブジェクトと呼ばれます。プロセス内で使用するために、オブジェクトグループを処理し、データを組み立てます。名前付きプロセスオブジェクトは、返された情報により効率的にアクセスするためにも役立ちます。プロセスオブジェクトを使用して、複数のプロセスで再利用可能な共通のオブジェクトを作成できます。

プロセスオブジェクトは次の 2 種類の方法で定義できます。

1. アプリケーション内で使用する目的で、プロセスオブジェクトをスタンドアロンとして定義します。
2. サービスから返されたデータを処理する目的で、サービスコネクタ内で定義します。

サービスコネクタ内で生成されたプロセスオブジェクトは、そのサービスコネクタ内でのみ使用できます。スタンドアロンプロセスオブジェクトは、すべてのサービスコネクタ内で使用できます。どちらもプロセス内で同様に使用されます。

プロセスオブジェクトエディタを使用して、スタンドアロンプロセスオブジェクトを定義できます。その後、整数型、テキスト型、電話番号型などの組み込みのデータ型と同様に使用できるようになります。サービスコネクタエディタでは、サービスから使用できるデータ、アクション、バインディング、ここで指定するその他のプロパティに基づいて、プロセスオブジェクトを生成できます。

## プロセスオブジェクト作成オプション

プロセスオブジェクトは、次の方法で作成できます。

### プロセスオブジェクトの手動作成

単純なプロセスオブジェクトを手動で作成するには、フィールドを追加し、フィールドプロパティを設定します。

### WSDL ファイルおよび XSD ファイルからのプロセスオブジェクトの作成

1 つ以上の WSDL ファイルおよび XSD ファイルをインポートすることで、複数の大きな階層プロセスオブジェクトを簡単に同時に作成できます。

### Swagger ファイルからのプロセスオブジェクトの作成

1 つ以上の Swagger JSON ファイルまたは YAML ファイルをインポートすることで、複数の大きな階層プロセスオブジェクトを簡単に同時に作成できます。

### OpenAPI 3.0 ファイルからのプロセスオブジェクトの作成

1 つ以上の OpenAPI 3.0 JSON ファイルまたは YAML ファイルをインポートすることで、複数の大きな階層プロセスオブジェクトを簡単に同時に作成できます。



## JSON ファイルからのプロセスオブジェクトの作成

1つ以上の JSON ファイルをインポートすることで、JSON ペイロードで使用されるデータ型をカバーする複数の大規模な階層プロセスオブジェクトを簡単に作成できます。

# プロセスオブジェクトの手動作成

単純なプロセスオブジェクトを手動で作成するには、フィールドを追加し、フィールドプロパティを設定します。

1. アプリケーション統合で、**[新規]** > **[プロセスオブジェクト]** > **[プロセスオブジェクト]** > **[作成]** の順にクリックします。

プロセスオブジェクトエディタが表示されます。

次の図は、プロセスオブジェクトエディタを示しています。

Name*	Label	Type	Required	Nullable
-------	-------	------	----------	----------

2. プロセスオブジェクトの名前を指定します。  
名前は英字で始まる必要があり、英数字、マルチバイト文字、アンダースコア ( \_ )、およびハイフン ( - ) のみを含めることができます。120 文字を超える名前は使用できません。これは必須フィールドです。
3. **[参照]** をクリックし、プロセスオブジェクトを保存する場所を指定します。
4. 必要に応じて、プロセスオブジェクトの説明を入力します。
5. プロセスオブジェクトにフィールドを追加するには、以下の手順を実行します。
  - a. フィールドの名前を入力します。  
名前は先頭の文字を英字にする必要があり、名前には英数字とハイフン ( - ) のみを含めることができます。
  - b. 必要に応じて、フィールドのラベルを入力します。
  - c. **[型]** リストから、フィールドの適切なデータ型を選択します。  
別のプロセスオブジェクトを参照する場合は、型に **[参照]** を選択します。  
名前を入力するには、タイプとして **[テキスト]** を選択します。
  - d. **[名前]** フィールドを展開し、**[長さ]** フィールドにテキストの長さを入力します。  
日付データ型の場合は日付範囲、日時データ型の場合は日時範囲、時間データ型の場合は時間範囲、数値データ型の場合は最小値と最大値、また、整数データ型の場合は最小値と最大値を入力できません。

- e. **【形式】** フィールドにテキストの形式を入力します。
  - f. フィールドをプロセスオブジェクトに追加する場合は、**【フィールドの追加】** をクリックします。  
型に **【オブジェクトリスト】** または **【参照】** を選択した場合は、参照するプロセスオブジェクトを参照して選択します。**【検索】** ボックスにプロセスオブジェクト名を入力して、プロセスオブジェクトを検索することもできます。  
**【参照先オブジェクトの表示】** をクリックすると、プロセスオブジェクトエディタ自体から参照先プロセスオブジェクトを表示できます。参照先プロセスオブジェクトが左側のナビゲーションバーの新しいタブに読み取り専用モードで開きます。  
型に **【選択リスト】** または **【複数選択リスト】** を選択した場合、1 つ以上のオプションをカンマで区切って入力します。
  - g. フィールドを必須フィールドとして設定するには、**【必須】** を選択します。  
必須フィールドは、要求ペイロードに含まれている必要があります。値を含めることも、空にすることも、NULL にすることもできます。
  - h. フィールドが NULL 値を受け入れないようにする場合は、**【NULL 可能】** チェックボックスをクリアします。デフォルトでは、このチェックボックスはオンです。
6. プロセスオブジェクトに必須フィールドを設定します。

次の図は、[ID]、[名前]、[住所] の 3 つのフィールドを含む [従業員] プロセスオブジェクトを示しています。

The screenshot shows the configuration interface for the 'EmployeeProcessObject'. At the top, there is a 'Properties' section with fields for 'Name' (EmployeeProcessObject), 'Locations' (Default), and 'Description'. Below this is the 'Fields' section, which includes a table of defined fields and a 'Reference' section for the 'Address' field.

Name*	Label	Type	Required	Nullable
ID	ID	Number	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Name	name	Text	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Address	Address	Reference	<input type="checkbox"/>	<input checked="" type="checkbox"/>

The 'Reference' section for the 'Address' field shows a 'View Referenced Process Object' link and a dropdown menu with 'Default > Address' selected.

フィールドを並べ替えるには、フィールドを含む行を必要な位置にドラッグアンドドロップします。

フィールドを削除するには、そのフィールドがある行にカーソルを合わせ、**【削除】** アイコンをクリックします。

7. フィールド値を編集する場合は、フィールドを含む行の任意の場所をクリックします。  
行のすべてのフィールドが編集可能になります。
8. **【保存】** をクリックして、プロセスオブジェクトを保存します。

## プロセスでプロセスオブジェクトを使用するためのルールおよびガイドライン

プロセスでプロセスオブジェクトを使用する際には、次のルールとガイドラインを考慮してください。

- 必須フィールドをアプリケーション統合プロセスの入力とするプロセスオブジェクトを使用する場合は、必須フィールドが要求ペイロードに存在することを確認する必要があります。存在しない場合、プロセスを実行すると、フィールドがないためにエラーが発生します。

例えば、プロセスオブジェクトで、ID フィールドを必須として設定したものの、次のサンプルに示すように、このフィールドが要求ペイロードに存在しないとします。

```
<root xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Employee>
    <Name xsi:nil="true"></Name>
    <Address xsi:nil="true"></Address>
  </Employee>
</root>
```

プロセスを実行すると、以下のようなエラーメッセージが表示されます。

```
<root>
  <code>400</code>
  <details>
    <missing>[ID]</missing>
  </details>
  <message>The parameter validation failed</message>
</root>
```

- プロセスオブジェクトフィールドを NULL 可能として設定すると、有効な値を含めることも、NULL にすることもできます。NULL 不可のフィールドを NULL として設定することはできません。

JSON 形式と XML 形式では、アプリケーション統合プロセスの入力として使用されるプロセスオブジェクトフィールドの NULL 値は異なる方法で処理されます。

JSON 形式では、フィールド値として NULL 値を直接指定できます。

例: {"Name":null}

XML は NULL 値をサポートしていません。したがって、XML 形式では、フィールドが NULL 可能として設定されていて、NULL 値を渡す場合は、フィールド要素で xsi:nil 属性を "true" として設定し、次のサンプルに示すように、ルート要素に <xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"> を含める必要があります。

```
<root xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Employee>
    <ID>1234</ID>
    <Name xsi:nil="true"></Name>
    <Address xsi:nil="true"></Address>
  </Employee>
</root>
```

## WSDL ファイルおよび XSD ファイルからのプロセスオブジェクトの作成

1 つ以上の WSDL ファイルおよび XSD ファイルをインポートすることで、複数の大きな階層プロセスオブジェクトを簡単に同時に作成できます。

**[WSDL ファイルや XSD ファイルからの新規プロセスオブジェクトの作成]** ウィザードを使用して、プロセスオブジェクトを WSDL ファイルと XSD ファイルから作成します。ソースを指定すると、アプリケーション統合ではソースが解析され、リソースがロードされて、XSD スキーマが生成されます。XSD スキーマに基づいて、

アプリケーション統合ではプロセスオブジェクトが生成されます。プロセスオブジェクトを保存する場所などのインポート設定を指定し、必要に応じてプロセスオブジェクトに使用されるタグを指定できます。

プロセスオブジェクト作成中にエラーが発生した場合、ウィザードの該当するステップの【関連ログ】タブにエラーが表示されます。ログを確認して対処できます。ログをダウンロードすることもできます。特定のステップでログをダウンロードすると、ログにはそれより前のすべてのステップに関する累積ログ情報も表示されます。

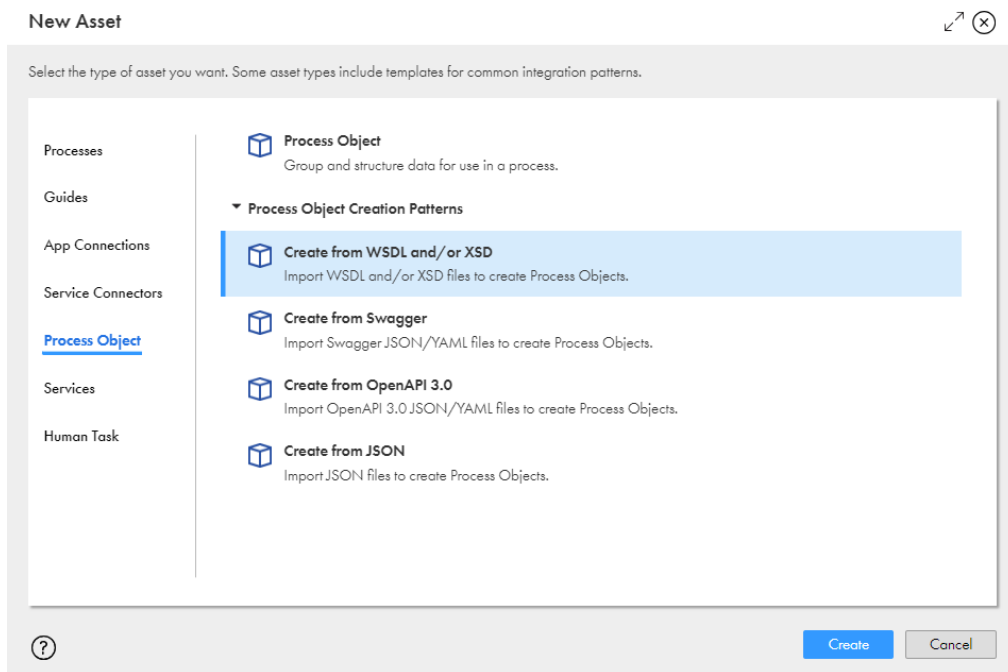
次のビデオは、WSDL ファイルをインポートしてプロセスオブジェクトを作成する方法を示しています。

<https://knowledge.informatica.com/s/article/DOC-18508>

1. アプリケーション統合で、【新規】をクリックします。

【新しいアセット】ダイアログボックスが表示されます。

次の図は、【新しいアセット】ダイアログボックスを示しています。



2. 左ペインの【プロセスオブジェクト】をクリックします。
3. 右ペインで、【プロセスオブジェクト作成パターン】リストを展開し、【WSDLまたはXSDからの作成】をクリックしてから、【作成】をクリックします。

【WSDL ファイルや XSD ファイルからの新規プロセスオブジェクト】ダイアログボックスが表示されます。

次の図は、【WSDL ファイルや XSD ファイルからの新規プロセスオブジェクト】ダイアログボックスを示しています。

## New Process Objects Creation from WSDL and/or XSD Files



1 Select Source   2 Load Resources   3 Generate Schemas  
4 Generate Objects   5 Import Preferences   6 Import Objects

WSDL and/or XSD Files

Source: \*    File    Zip (multiple files)    URL

Select File: \*     

4. **【ソース】** フィールドで、次のうちいずれかの値を選択します。

オプション	説明
ファイル	<p><b>【ファイル】</b> を選択し、ローカルシステムから型が定義されている単一の WSDL ソースファイルまたは XSD ソースファイルを指定します。HTTP リソースを指定することもできます。<b>【ファイルの選択】</b> を選択して、WSDL ソースファイルまたは XSD ソースファイルを選択します。</p> <p>単一ファイルの最大許容ファイルサイズは 5 MB です。 デフォルトは <b>【ファイル】</b> です。</p>
Zip (複数ファイル)	<p><b>【Zip】</b> を選択して、型がローカルシステムから zip ファイル形式で定義されている複数の WSDL ソースファイルまたは XSD ソースファイルを指定します。<b>【ファイルを選択】</b> をクリックして、zip ファイルを選択します。</p> <p>zip ファイルの最大許容ファイルサイズは 3 MB です。</p>
URL	<p><b>【URL】</b> を選択し、型が定義されている WSDL ソースファイルまたは XSD ソースファイルを含む URL を指定します。<b>【URL】</b> フィールドと <b>【認証の使用】</b> フィールドが表示されます。</p> <p><b>【URL】</b> フィールドに、ソース URL を入力します。</p> <p><b>【認証の使用】</b> オプションを選択して、ソース URL にアクセスするためのユーザー名とパスワードを指定します。</p>

5. **【次へ】** をクリックします。

アプリケーション統合では指定されたソースファイルまたは URL が解析され、**【リソースのロード】** タブにリソース詳細が表示されます。

次の図は、リソース詳細が表示された **【リソースのロード】** タブを示しています。

## New Process Objects Creation from WSDL and/or XSD Files



① Select Source   ② Load Resources   ③ Generate Schemas  
④ Generate Objects   ⑤ Import Preferences   ⑥ Import Objects

Files Found (3)		Related Logs			
File Name	File Path	Type	Status	Comments	
http://schemas.x...		XSD	✓ Valid	File type supported.	
ndfdXML.wsdl	https://graphical....	WSDL	✓ Valid	File type supported.	
http://schemas.x...		XSD	✓ Valid	File type supported.	

6. リソース詳細を確認します。  
以下の表に、【リソースのロード】タブに表示される列を示します。

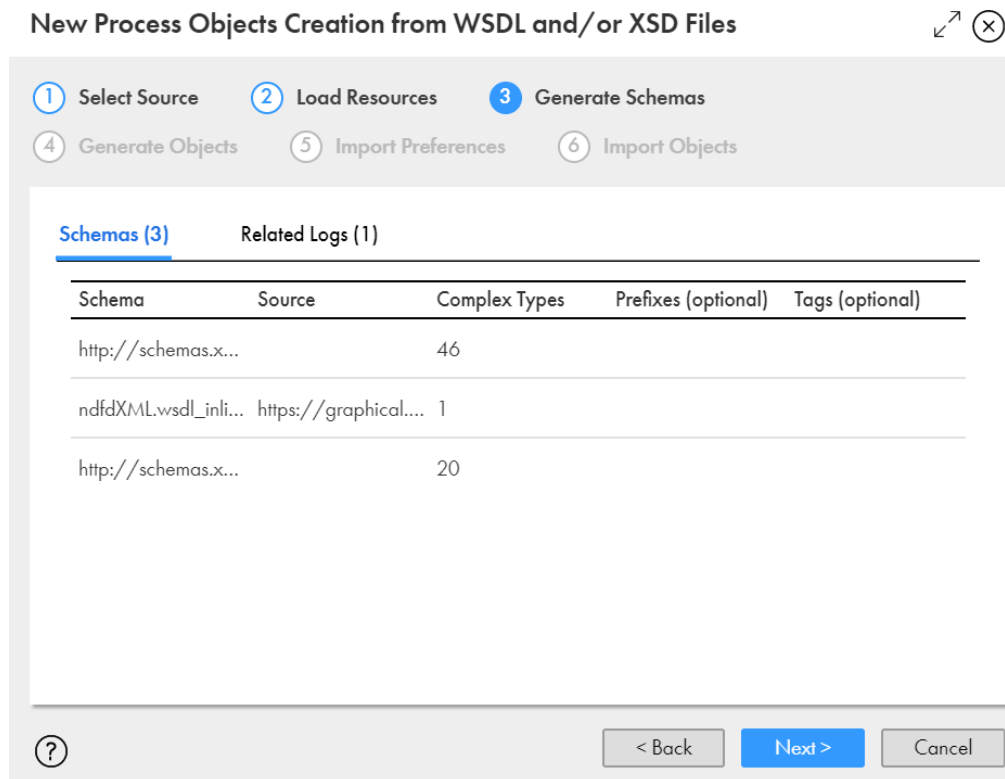
カラム名	説明
ファイル名	リソースのファイル名。
ファイルパス	ソース内の各リソースの URL または関連する場所。
タイプ	リソースのファイルタイプ。次のいずれかの値を表示します。 - XSD - WSDL
ステータス	リソースのステータス。次のいずれかの値を表示します。 - <b>有効</b> 。ファイルタイプがサポートされ、ファイルコンテンツが有効であることを示します。 - <b>無効</b> 。ファイルタイプがサポートされていないか、ファイルコンテンツが無効であることを示します。例えば、txt ファイルの場合、ファイルタイプがサポートされていないため、ステータスは【無効】と表示されます。
コメント	ファイルタイプがサポートされているかどうかを示します。

解析エラーがある場合、アプリケーション統合では【関連ログ】タブにエラー詳細が表示されます。エラーを修正してやり直してください。

7. 【次へ】をクリックします。

アプリケーション統合では、作成されるプロセスオブジェクトに基づいてリソースが XSD スキーマに変換されます。**【スキーマの生成】** タブにスキーマ詳細が表示されます。

次の図は、スキーマ詳細が表示された **【スキーマの生成】** タブを示しています。



8. スキーマ詳細を確認します。

以下の表に、**【スキーマの生成】** タブに表示される列を示します。

カラム名	説明
スキーマ	生成された XSD スキーマファイルの名前。
ソース	XSD スキーマファイルが生成されたソース。
複合型	XSD スキーマごとに作成される複合型の数。

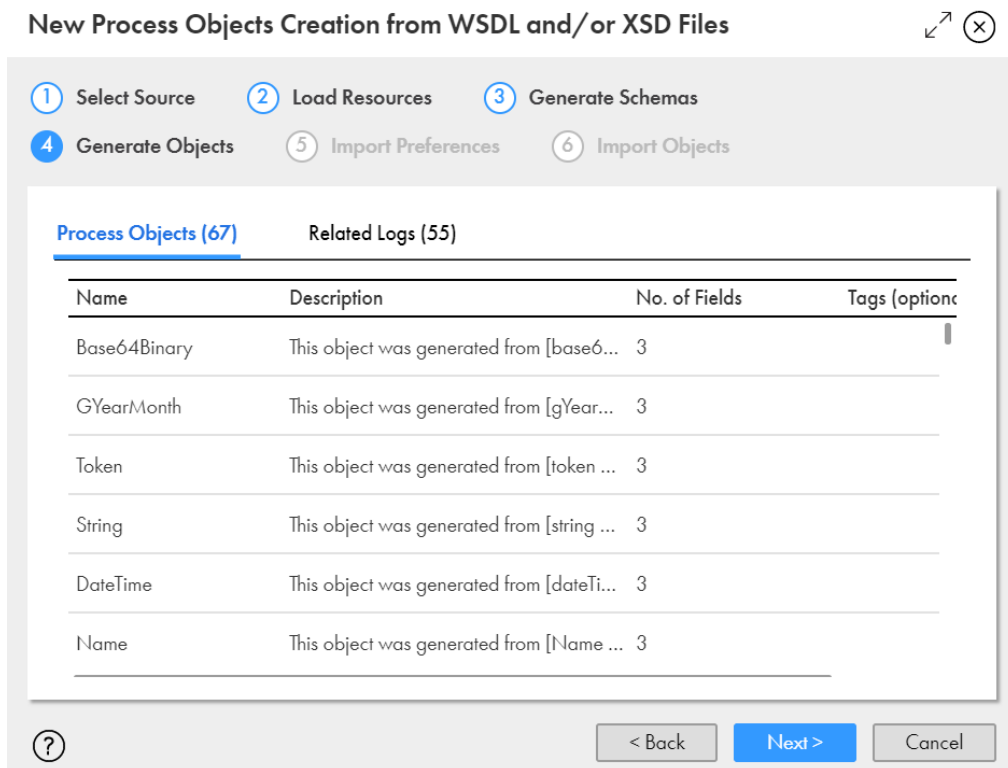
カラム名	説明
プレフィックス	<p>XSD スキーマに使用されるプレフィックス。</p> <p>プレフィックスを使用して、スキーマから作成されるプロセスオブジェクトを簡単に特定できます。例えば、Salesforce スキーマから作成されるすべてのプロセスオブジェクトに対しプレフィックス Salesforce を追加とします。</p> <p>スキーマに対しプレフィックスを指定するには、スキーマが含まれている行に対し【<b>プレフィックス</b>】列をダブルクリックし、プレフィックスを入力します。スキーマから作成されるすべてのプロセスオブジェクトでは、このフィールドで指定したものと同一プレフィックスが使用されます。</p>
タグ	<p>XSD スキーマに使用されるタグ。</p> <p>タグを使用して、関連するスキーマをまとめてグループ化することができます。例えば、財務リソースを含むすべてのスキーマをタグ Finance にグループ化とします。</p> <p>スキーマに対しタグを指定するには、スキーマが含まれている行に対し【<b>タグ</b>】列をダブルクリックし、タグを入力します。スキーマから作成されるすべてのプロセスオブジェクトでは、このフィールドで指定したものと同一プレフィックスが使用されます。</p>

スキーマ生成エラーがある場合、アプリケーション統合では【**関連ログ**】タブにエラー詳細が表示されます。エラーを修正してやり直してください。

9. 【**次へ**】をクリックします。

アプリケーション統合では、XSD スキーマに指定したプレフィックスとタグが適用され、プロセスオブジェクトが XSD スキーマに基づいて生成されます。【**オブジェクトの生成**】タブにプロセスオブジェクト詳細が表示されます。

次の図は、プロセスオブジェクト詳細が表示された【**オブジェクトの生成**】タブを示しています。





10. プロセスオブジェクトの詳細を確認します。

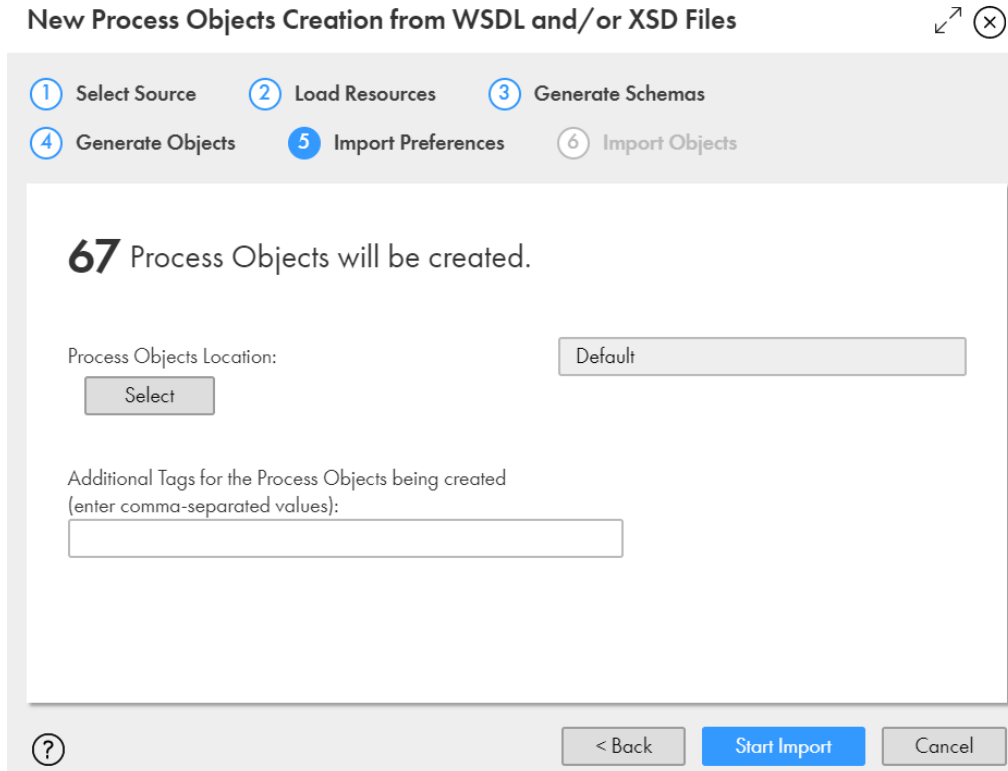
以下の表に、**【オブジェクトの生成】** タブに表示される列を示します。

カラム名	説明
名前	スキーマから生成されたプロセスオブジェクトの名前。
説明	プロセスオブジェクトが生成されたスキーマ。説明を編集できます。
いいえフィールド関連	プロセスオブジェクトに含まれている複合型の数。
タグ	プロセスオブジェクトに使用されるタグ。デフォルトで、プロセスオブジェクトタグは XSD スキーマに指定したタグから派生します。 プロセスオブジェクトレベルで別のタグを指定する場合はタグを編集できます。 プロセスオブジェクトのタグを編集するには、プロセスオブジェクトを含む行に対し <b>【タグ】</b> 列をダブルクリックし、タグを更新します

11. **【次へ】** をクリックします。

作成されるプロセスオブジェクトの数を表示する **【設定のインポート】** タブが表示されます。

次の図は、**【設定のインポート】** タブを示しています。

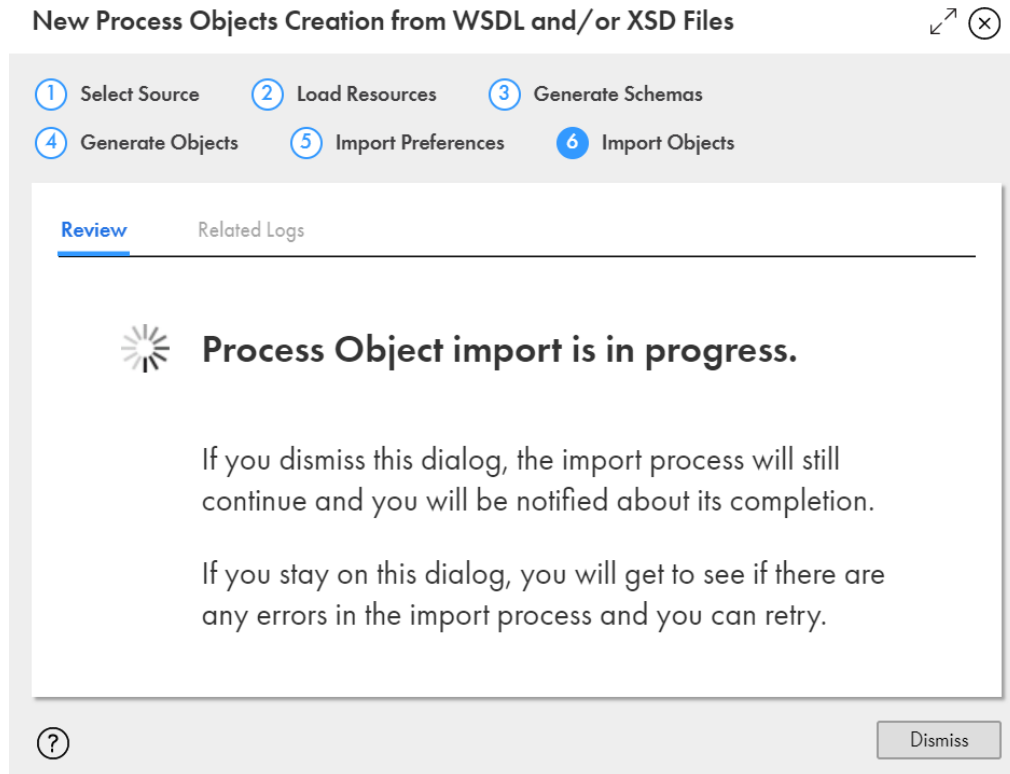


12. 必要に応じて、**【選択】** をクリックして、プロセスオブジェクトを保存する場所を指定します。  
デフォルトは、**【WSDL ファイルや XSD ファイルからの新規プロセスオブジェクト】** ウィザードを開いたときに選択された場所です。
13. 必要に応じて、プロセスオブジェクトに対し追加のタグをカンマ区切り値として指定します。

14. **【インポートの開始】** をクリックします。

アプリケーション統合では、プロセスオブジェクトのインポートが開始され、インポートステータスが表示されます。

次の図は、インポートステータスを示しています。



15. 以下のいずれかのアクションを実行します。

- ダイアログボックスを破棄し、プロセスオブジェクトのインポート完了時に通知を受けるには、**【破棄】** をクリックします。
- インポートエラーを表示し、修正してインポートを再試行するには、ダイアログボックスを開いたままにします。

プロセスオブジェクトのインポート後、アプリケーション統合ではインポートが成功または失敗したかどうかを示すステータスメッセージが表示されます。

インポートが失敗した場合、**【関連ログ】** タブを確認してエラーを修正し、プロセスオブジェクトのインポートを再試行します。**【ダウンロード】** アイコンをクリックして、ログ全体をダウンロードすることもできます。

次の図は、1つのプロセスオブジェクトがPOST操作に対する401エラーによってインポートされなかったことを示すサンプルログを示しています。

## New Process Objects Creation from WSDL and/or XSD Files



16. **【完了】** をクリックします。
17. プロセスオブジェクト生成に指定した場所へ移動し、プロセスオブジェクトを確認します。

## Swagger ファイルからのプロセスオブジェクトの作成

1 つ以上の Swagger JSON ファイルまたは YAML ファイルをインポートすることで、複数の大きな階層プロセスオブジェクトを簡単に同時に作成できます。

**【Swagger ファイルからの新規プロセスオブジェクト】** ウィザードを使用して、プロセスオブジェクトを Swagger ファイルから作成します。ソースを指定すると、アプリケーション統合ではソースが解析され、リソースがロードされて、XSD スキーマが生成されます。XSD スキーマに基づいて、アプリケーション統合ではプロセスオブジェクトが生成されます。プロセスオブジェクトを保存する場所などのインポート設定を指定し、必要に応じてプロセスオブジェクトに使用されるタグを指定できます。

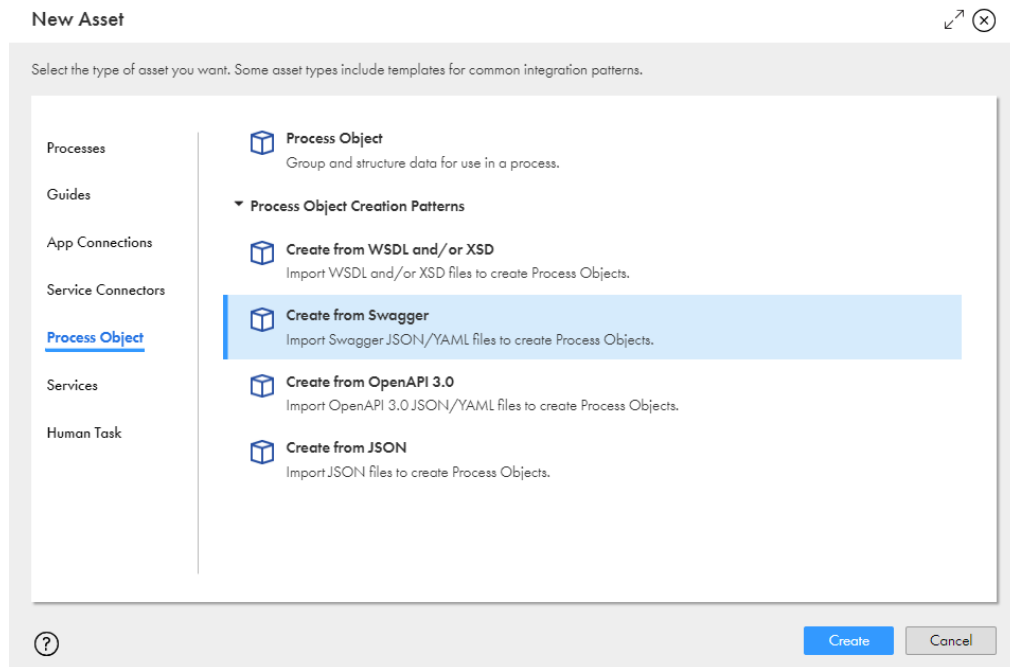
プロセスオブジェクト作成中にエラーが発生した場合、ウィザードの該当するステップの **【関連ログ】** タブにエラーが表示されます。ログを確認して対処できます。ログをダウンロードすることもできます。特定のステ

ップでログをダウンロードすると、ログにはそれより前のすべてのステップに関する累積ログ情報も表示されます。

1. アプリケーション統合で、**【新規】** をクリックします。

**【新しいアセット】** ダイアログボックスが表示されます。

次の図は、**【新しいアセット】** ダイアログボックスを示しています。

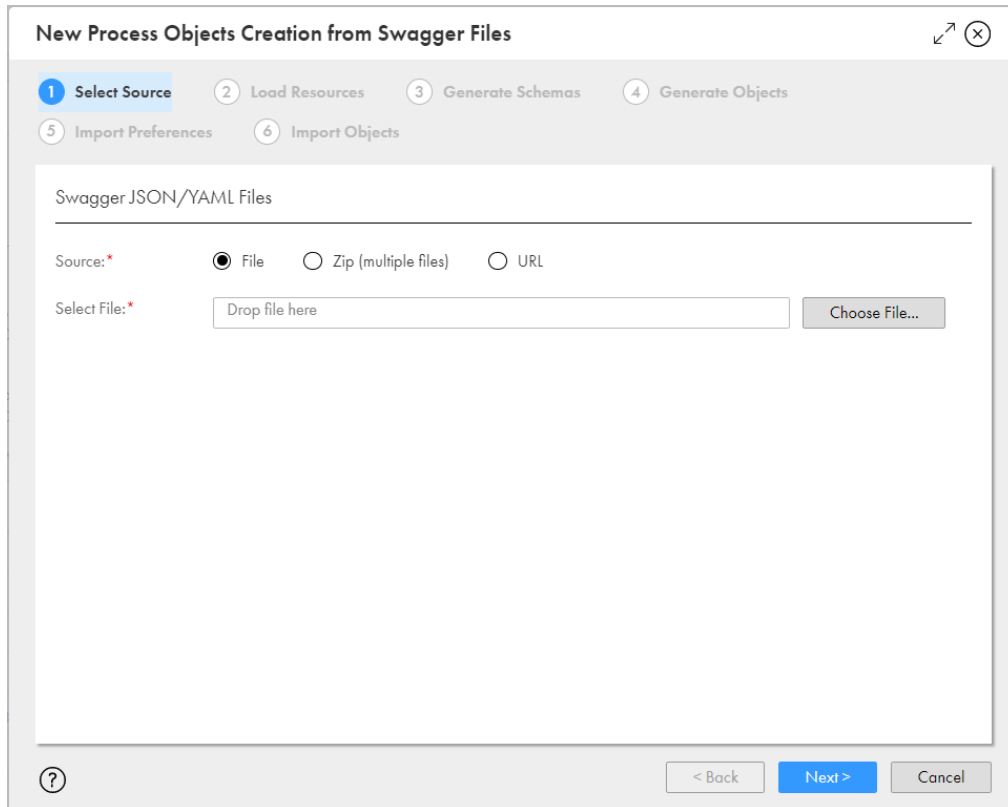


2. 左ペインの **【プロセスオブジェクト】** をクリックします。

3. 右ペインで、**【プロセスオブジェクト作成パターン】** リストを展開し、**【Swagger からの作成】** をクリックしてから、**【作成】** をクリックします。

**【Swagger ファイルからの新規プロセスオブジェクト】** ダイアログボックスが表示されます。

次の図は、**【Swagger ファイルからの新規プロセスオブジェクト】** ダイアログボックスを示しています。



4. **【ソース】** フィールドで、次のうちいずれかの値を選択します。

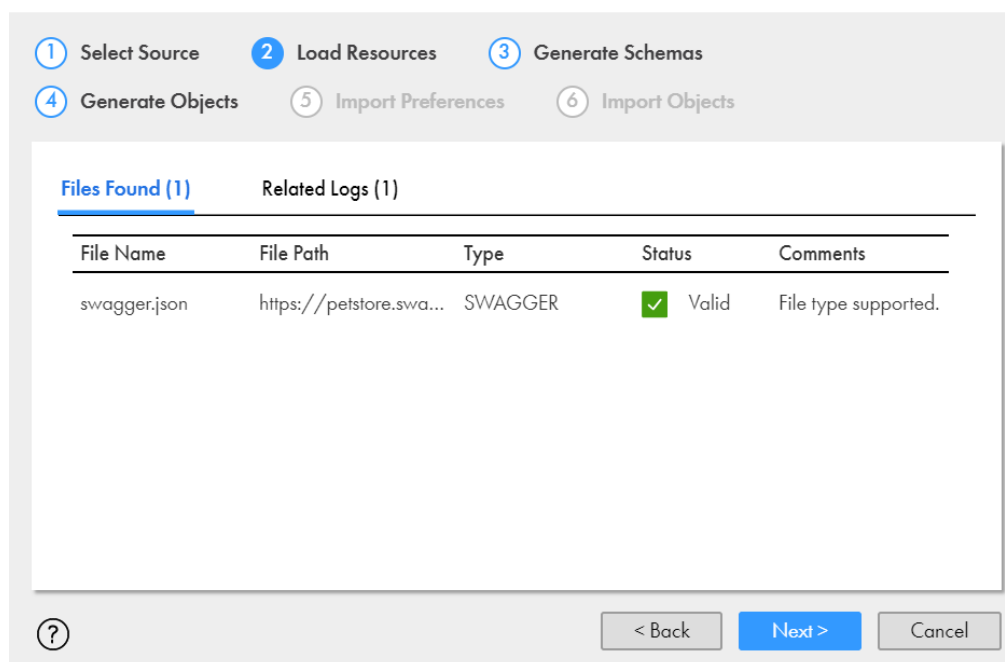
オプション	説明
ファイル	<p><b>【ファイル】</b> を選択し、ローカルシステムから単一の Swagger ソースファイルを指定します。<b>【ファイルを選択】</b> をクリックして、Swagger ソースファイルを選択します。単一ファイルの最大許容ファイルサイズは 5 MB です。デフォルトは <b>【ファイル】</b> です。</p>
Zip (複数ファイル)	<p><b>【Zip】</b> を選択して、ローカルシステムから zip ファイル形式の複数の Swagger ソースファイルを指定します。<b>【ファイルを選択】</b> をクリックして、zip ファイルを選択します。zip ファイルの最大許容ファイルサイズは 3 MB です。</p>
URL	<p><b>【URL】</b> を選択して、Swagger ソースファイルを含む URL を指定します。<b>【URL】</b> フィールドと <b>【認証の使用】</b> フィールドが表示されます。<b>【URL】</b> フィールドに、ソース URL を入力します。<b>【認証の使用】</b> オプションを選択して、ソース URL にアクセスするためのユーザー名とパスワードを指定します。</p>

5. **【次へ】** をクリックします。

アプリケーション統合では指定されたソースファイルまたは URL が解析され、**【リソースのロード】** タブにリソース詳細が表示されます。

次の図は、リソース詳細が表示された **【リソースのロード】** タブを示しています。

## New Process Objects Creation from Swagger Files



6. リソース詳細を確認します。  
以下の表に、【リソースのロード】タブに表示される列を示します。

カラム名	説明
ファイル名	リソースのファイル名。
ファイルパス	ソース内の各リソースの URL または関連する場所。
タイプ	リソースのファイルタイプ。値が【SWAGGER】と表示されます。
ステータス	リソースのステータス。次のいずれかの値を表示します。 - 有効。ファイルタイプがサポートされ、ファイルコンテンツが有効であることを示します。 - 無効。ファイルタイプがサポートされていないか、ファイルコンテンツが無効であることを示します。例えば、txt ファイルの場合、ファイルタイプがサポートされていないため、ステータスは【無効】と表示されます。
コメント	ファイルタイプがサポートされているかどうかを示します。

解析エラーがある場合、アプリケーション統合では【関連ログ】タブにエラー詳細が表示されます。エラーを修正してやり直してください。

7. 【次へ】をクリックします。  
アプリケーション統合では、作成されるプロセスオブジェクトに基づいてリソースが XSD スキーマに変換されます。【スキーマの生成】タブにスキーマ詳細が表示されます。  
次の図は、スキーマ詳細が表示された【スキーマの生成】タブを示しています。

## New Process Objects Creation from Swagger Files



① Select Source   ② Load Resources   ③ Generate Schemas  
④ Generate Objects   ⑤ Import Preferences   ⑥ Import Objects

**Schemas (1)**   Related Logs

Schema	Source	Complex Types	Prefixes (optional)	Tags (optional)
swagger.json.xsd	https://petstore.s...	6		

①   < Back   Next >   Cancel

8. スキーマ詳細を確認します。  
以下の表に、**[スキーマの生成]** タブに表示される列を示します。

カラム名	説明
スキーマ	生成された XSD スキーマファイルの名前。
ソース	XSD スキーマファイルが生成されたソース。
複合型	XSD スキーマごとに作成される複合型の数。

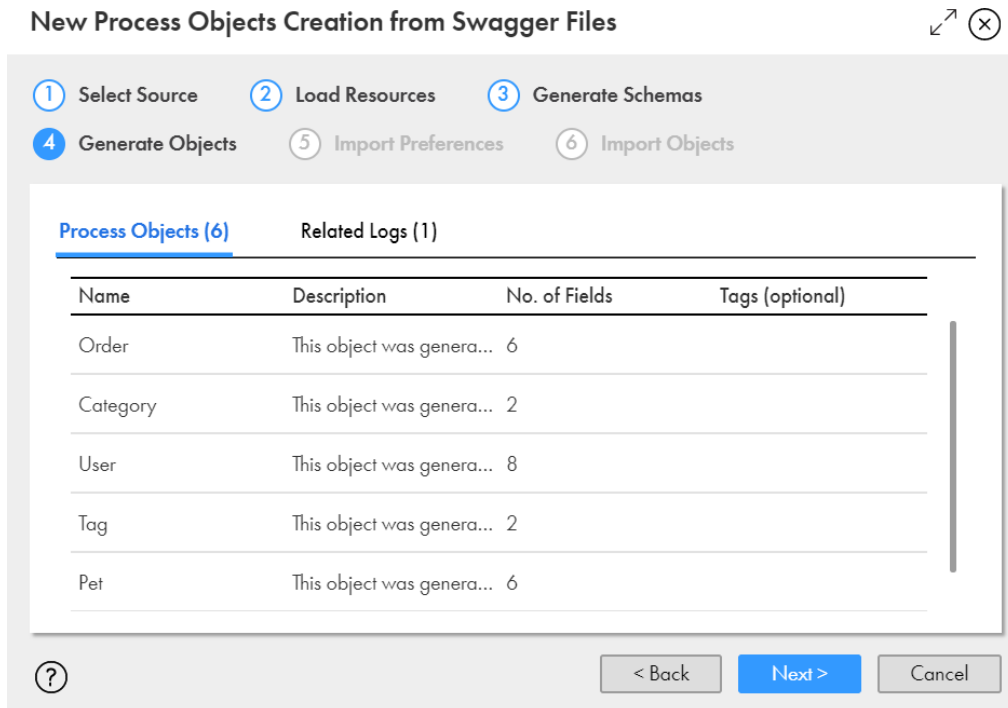
カラム名	説明
プレフィックス	<p>XSD スキーマに使用されるプレフィックス。</p> <p>プレフィックスを使用して、スキーマから作成されるプロセスオブジェクトを簡単に特定できます。例えば、Salesforce スキーマから作成されるすべてのプロセスオブジェクトに対しプレフィックス Salesforce を追加とします。</p> <p>スキーマに対しプレフィックスを指定するには、スキーマが含まれている行に対し【プレフィックス】列をダブルクリックし、プレフィックスを入力します。スキーマから作成されるすべてのプロセスオブジェクトでは、このフィールドで指定したものと同一プレフィックスが使用されます。</p>
タグ	<p>XSD スキーマに使用されるタグ。</p> <p>タグを使用して、関連するスキーマをまとめてグループ化することができます。例えば、財務リソースを含むすべてのスキーマをタグ Finance にグループ化とします。</p> <p>スキーマに対しタグを指定するには、スキーマが含まれている行に対し【タグ】列をダブルクリックし、タグを入力します。スキーマから作成されるすべてのプロセスオブジェクトでは、このフィールドで指定したものと同一プレフィックスが使用されます。</p>

スキーマ生成エラーがある場合、アプリケーション統合では【関連ログ】タブにエラー詳細が表示されます。エラーを修正してやり直してください。

9. 【次へ】をクリックします。

アプリケーション統合では、XSD スキーマに指定したプレフィックスとタグが適用され、プロセスオブジェクトが XSD スキーマに基づいて生成されます。【オブジェクトの生成】タブにプロセスオブジェクト詳細が表示されます。

次の図は、プロセスオブジェクト詳細が表示された【オブジェクトの生成】タブを示しています。



10. プロセスオブジェクトの詳細を確認します。



以下の表に、【オブジェクトの生成】 タブに表示される列を示します。

カラム名	説明
名前	スキーマから生成されたプロセスオブジェクトの名前。
説明	プロセスオブジェクトが生成されたスキーマ。説明を編集できます。
いいえフィールド関連	プロセスオブジェクトに含まれている複合型の数。
タグ	プロセスオブジェクトに使用されるタグ。デフォルトで、プロセスオブジェクトタグは XSD スキーマに指定したタグから派生します。 プロセスオブジェクトレベルで別のタグを指定する場合はタグを編集できます。 プロセスオブジェクトのタグを編集するには、プロセスオブジェクトを含む行に対し【タグ】列をダブルクリックし、タグを更新します

11. 【次へ】 をクリックします。

作成されるプロセスオブジェクトの数を表示する【設定のインポート】タブが表示されます。

次の図は、【設定のインポート】タブを示しています。

New Process Objects Creation from Swagger Files

① Select Source   ② Load Resources   ③ Generate Schemas  
④ Generate Objects   ⑤ Import Preferences   ⑥ Import Objects

6 Process Objects will be created.

Process Objects Location:

Additional Tags for the Process Objects being created  
(enter comma-separated values):

12. 必要に応じて、【選択】 をクリックして、プロセスオブジェクトを保存する場所を指定します。

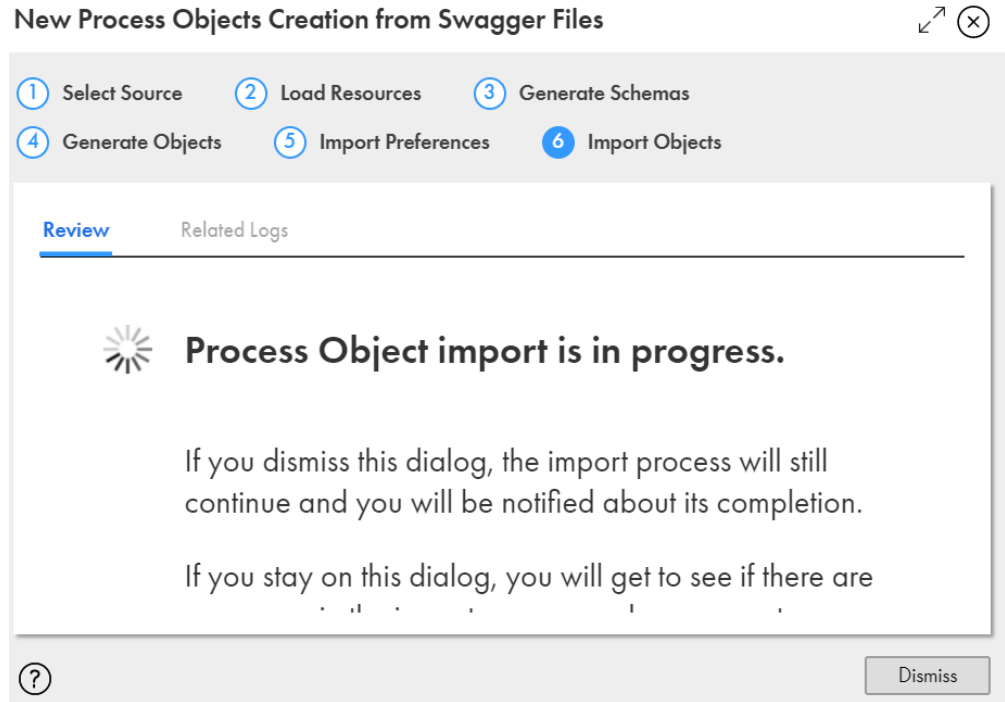
デフォルトは、Swagger ファイルからの新規プロセスオブジェクトウィザードを開いたときに選択された場所です。

13. 必要に応じて、プロセスオブジェクトに対し追加のタグをカンマ区切り値として指定します。

14. 【インポートの開始】 をクリックします。

アプリケーション統合では、プロセスオブジェクトのインポートが開始され、インポートステータスが表示されます。

次の図は、インポートステータスを示しています。



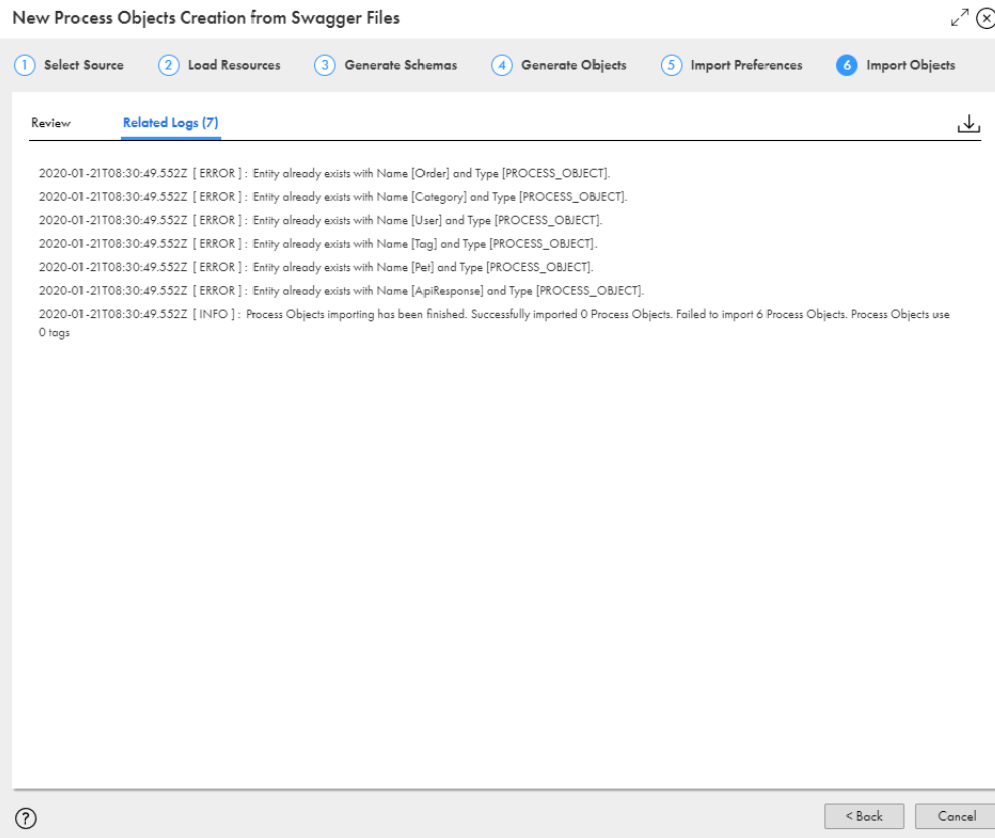
15. 以下のいずれかのアクションを実行します。

- ダイアログボックスを破棄し、プロセスオブジェクトのインポート完了時に通知を受けるには、**【破棄】** をクリックします。
- インポートエラーを表示し、修正してインポートを再試行するには、ダイアログボックスを開いたままにします。

プロセスオブジェクトのインポート後、アプリケーション統合ではインポートが成功または失敗したかどうかを示すステータスメッセージが表示されます。

インポートが失敗した場合、**【関連ログ】** タブを確認してエラーを修正し、プロセスオブジェクトのインポートを再試行します。**【ダウンロード】** アイコンをクリックして、ログ全体をダウンロードすることもできます。

次の図は、6つのプロセスオブジェクトが指定された場所にすでに存在するためインポートされなかったことを示すサンプルログを示しています。



16. **【完了】** をクリックします。
17. プロセスオブジェクト生成に指定した場所に移動し、プロセスオブジェクトを確認します。

## OpenAPI 3.0 ファイルからのプロセスオブジェクトの作成

1つ以上の OpenAPI 3.0 JSON ファイルまたは YAML ファイルをインポートすることで、複数の大きな階層プロセスオブジェクトを簡単に同時に作成できます。

**【OpenAPI 3.0 ファイルからの新規プロセスオブジェクト作成】** ウィザードを使用して、OpenAPI 3.0 ファイルからプロセスオブジェクトを作成します。ソースを指定すると、アプリケーション統合ではソースが解析され、リソースがロードされて、XSD スキーマが生成されます。XSD スキーマに基づいて、アプリケーション統合ではプロセスオブジェクトが生成されます。プロセスオブジェクトを保存する場所などのインポート設定を指定し、必要に応じてプロセスオブジェクトに使用されるタグを指定できます。

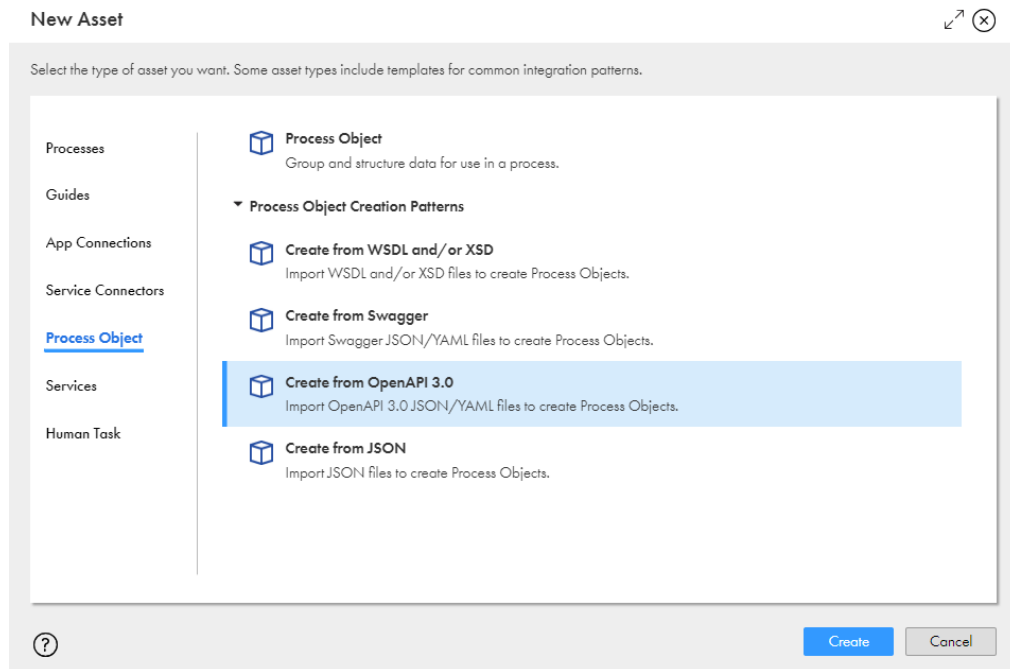
プロセスオブジェクト作成中にエラーが発生した場合、ウィザードの該当するステップの **【関連ログ】** タブにエラーが表示されます。ログを確認して対処できます。ログをダウンロードすることもできます。特定のステ

ップでログをダウンロードすると、ログにはそれより前のすべてのステップに関する累積ログ情報も表示されます。

1. アプリケーション統合で、**【新規】** をクリックします。

**【新しいアセット】** ダイアログボックスが表示されます。

次の図は、**【新しいアセット】** ダイアログボックスを示しています。

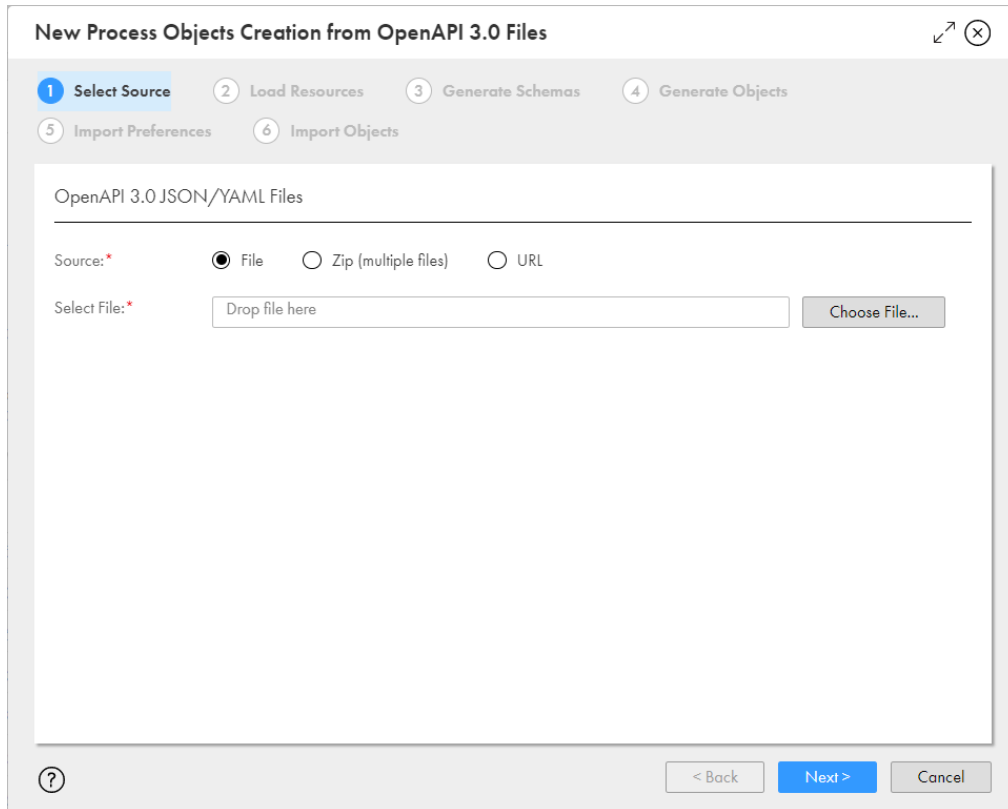


2. 左ペインの **【プロセスオブジェクト】** をクリックします。

3. 右ペインで、**【プロセスオブジェクト作成パターン】** リストを展開し、**【OpenAPI 3.0 から作成】** をクリックしてから、**【作成】** をクリックします。

**【OpenAPI 3.0 ファイルからの新規プロセスオブジェクト作成】** ダイアログボックスが表示されます。

次の図は、**【OpenAPI 3.0 ファイルからの新規プロセスオブジェクト作成】** ダイアログボックスを示しています。



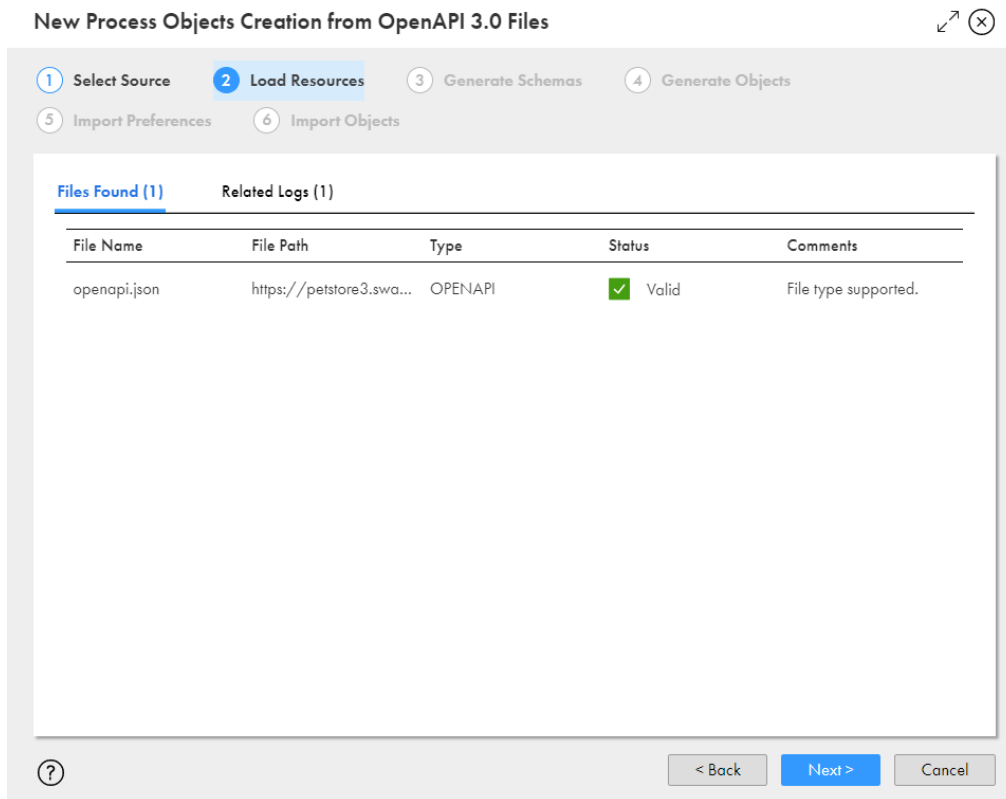
4. **【ソース】** フィールドで、次のうちいずれかの値を選択します。

オプション	説明
ファイル	<p><b>【ファイル】</b> を選択し、型が定義されている単一の OpenAPI 3.0 ソースファイルをローカルシステムから指定します。HTTP リソースを指定することもできます。<b>【ファイルを選択】</b> をクリックして、OpenAPI 3.0 ソースファイルを選択します。</p> <p>単一ファイルの最大許容ファイルサイズは 5 MB です。</p> <p>デフォルトは <b>【ファイル】</b> です。</p>
Zip (複数ファイル)	<p><b>【Zip】</b> を選択して、zip ファイル形式で型が定義されている複数の OpenAPI 3.0 ソースファイルをローカルシステムから指定します。<b>【ファイルを選択】</b> をクリックして、zip ファイルを選択します。</p> <p>zip ファイルの最大許容ファイルサイズは 3 MB です。</p>
URL	<p><b>【URL】</b> を選択し、型が定義されている OpenAPI 3.0 ソースファイルを含む URL を指定します。<b>【URL】</b> フィールドと <b>【認証の使用】</b> フィールドが表示されます。</p> <p><b>【URL】</b> フィールドに、ソース URL を入力します。</p> <p><b>【認証の使用】</b> オプションを選択して、ソース URL にアクセスするためのユーザー名とパスワードを指定します。</p>

5. **【次へ】** をクリックします。

アプリケーション統合では指定されたソースファイルまたは URL が解析され、**【リソースのロード】** タブにリソース詳細が表示されます。

次の図は、リソース詳細が表示された **【リソースのロード】** タブを示しています。



6. リソース詳細を確認します。  
以下の表に、**【リソースのロード】** タブに表示される列を示します。

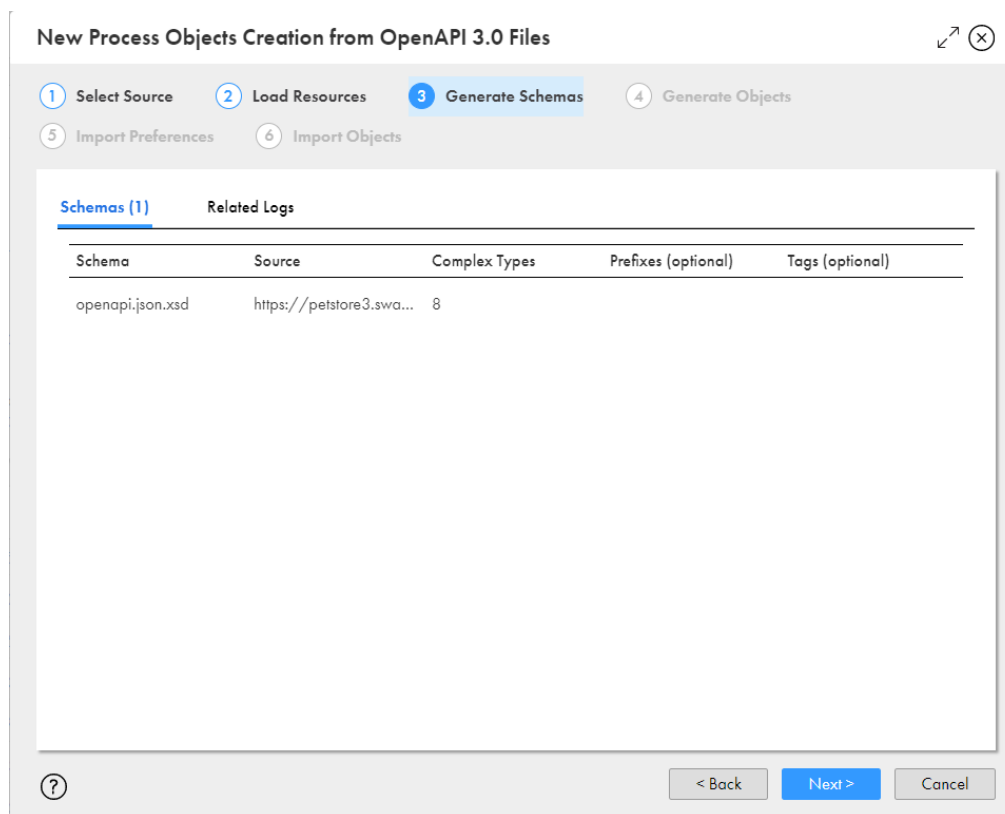
カラム名	説明
ファイル名	リソースのファイル名。
ファイルパス	ソース内の各リソースの URL または関連する場所。
タイプ	リソースのファイルタイプ。値が <b>【OPENAPI】</b> のように表示されます。
ステータス	リソースのステータス。次のいずれかの値を表示します。 <ul style="list-style-type: none"> <li>- <b>有効</b>。ファイルタイプがサポートされ、ファイルコンテンツが有効であることを示します。</li> <li>- <b>無効</b>。ファイルタイプがサポートされていないか、ファイルコンテンツが無効であることを示します。例えば、txt ファイルの場合、ファイルタイプがサポートされていないため、ステータスは <b>【無効】</b> と表示されます。</li> </ul>
コメント	ファイルタイプがサポートされているかどうかを示します。

解析エラーがある場合、アプリケーション統合では **【関連ログ】** タブにエラー詳細が表示されます。エラーを修正してやり直してください。

7. **【次へ】** をクリックします。

アプリケーション統合では、作成されるプロセスオブジェクトに基づいてリソースが XSD スキーマに変換されます。【スキーマの生成】タブにスキーマ詳細が表示されます。

次の図は、スキーマ詳細が表示された【スキーマの生成】タブを示しています。



8. スキーマ詳細を確認します。

以下の表に、【スキーマの生成】タブに表示される列を示します。

カラム名	説明
スキーマ	生成された XSD スキーマファイルの名前。
ソース	XSD スキーマファイルが生成されたソース。
複合型	XSD スキーマごとに作成される複合型の数。

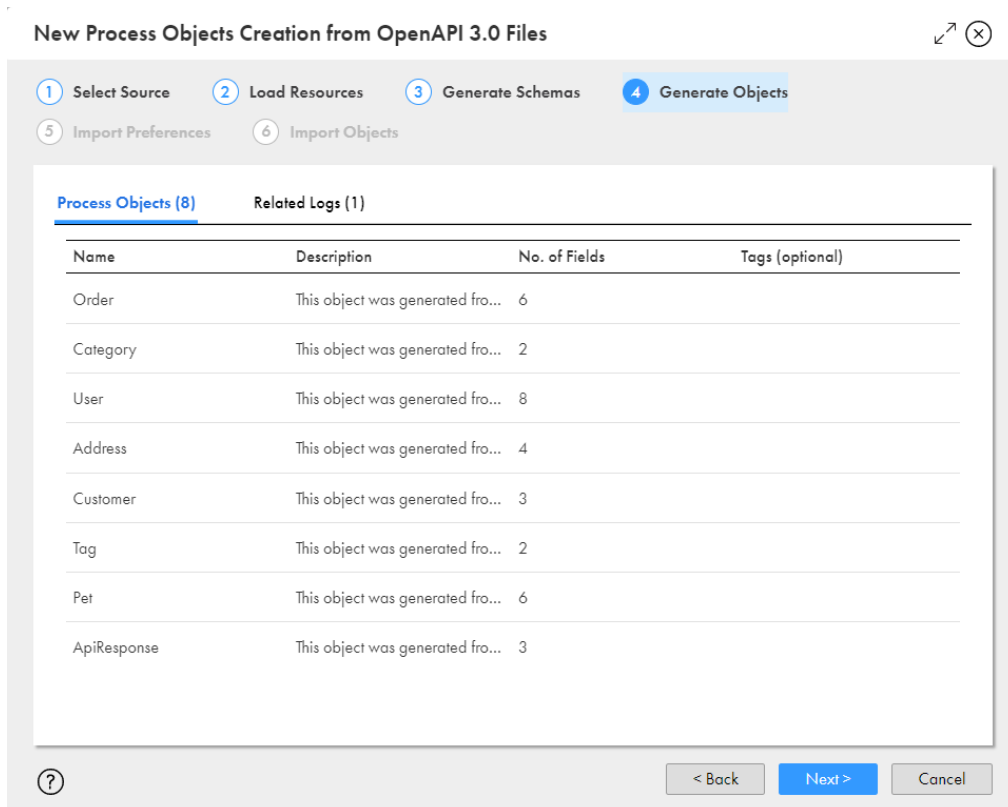
カラム名	説明
プレフィックス	<p>XSD スキーマに使用されるプレフィックス。</p> <p>プレフィックスを使用して、スキーマから作成されるプロセスオブジェクトを簡単に特定できます。例えば、Salesforce スキーマから作成されるすべてのプロセスオブジェクトに対しプレフィックス Salesforce を追加とします。</p> <p>スキーマに対しプレフィックスを指定するには、スキーマが含まれている行に対し【プレフィックス】列をダブルクリックし、プレフィックスを入力します。スキーマから作成されるすべてのプロセスオブジェクトでは、このフィールドで指定したものと同一プレフィックスが使用されます。</p>
タグ	<p>XSD スキーマに使用されるタグ。</p> <p>タグを使用して、関連するスキーマをまとめてグループ化することができます。例えば、財務リソースを含むすべてのスキーマをタグ Finance にグループ化とします。</p> <p>スキーマに対しタグを指定するには、スキーマが含まれている行に対し【タグ】列をダブルクリックし、タグを入力します。スキーマから作成されるすべてのプロセスオブジェクトでは、このフィールドで指定したものと同一プレフィックスが使用されます。</p>

スキーマ生成エラーがある場合、アプリケーション統合では【関連ログ】タブにエラー詳細が表示されます。エラーを修正してやり直してください。

9. 【次へ】をクリックします。

アプリケーション統合では、XSD スキーマに指定したプレフィックスとタグが適用され、プロセスオブジェクトが XSD スキーマに基づいて生成されます。【オブジェクトの生成】タブにプロセスオブジェクト詳細が表示されます。

次の図は、プロセスオブジェクト詳細が表示された【オブジェクトの生成】タブを示しています。





10. プロセスオブジェクトの詳細を確認します。  
以下の表に、**【オブジェクトの生成】** タブに表示される列を示します。

カラム名	説明
名前	スキーマから生成されたプロセスオブジェクトの名前。
説明	プロセスオブジェクトが生成されたスキーマ。説明を編集できます。
いいえフィールド関連	プロセスオブジェクトに含まれている複合型の数。
タグ	プロセスオブジェクトに使用されるタグ。デフォルトで、プロセスオブジェクトタグは XSD スキーマに指定したタグから派生します。 プロセスオブジェクトレベルで別のタグを指定する場合はタグを編集できます。 プロセスオブジェクトのタグを編集するには、プロセスオブジェクトを含む行に対し <b>【タグ】</b> 列をダブルクリックし、タグを更新します

11. **【次へ】** をクリックします。  
作成されるプロセスオブジェクトの数を表示する **【設定のインポート】** タブが表示されます。  
次の図は、**【設定のインポート】** タブを示しています。

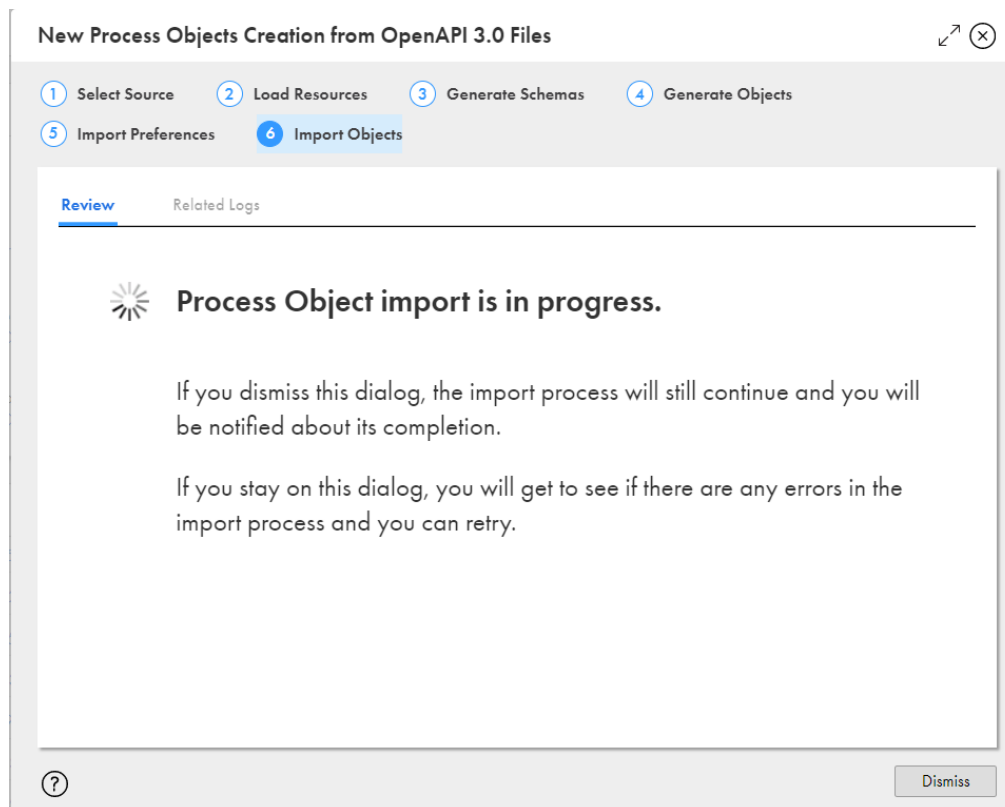
The screenshot shows a wizard window titled "New Process Objects Creation from OpenAPI 3.0 Files". The progress bar at the top indicates the current step is "5 Import Preferences". The main content area displays "8 Process Objects will be created." Below this, there is a "Process Objects Location:" label with a text input field containing "Default" and a "Select" button. Underneath, there is a label "Additional Tags for the Process Objects being created (enter comma-separated values):" followed by an empty text input field. At the bottom of the window, there are three buttons: "< Back", "Start Import", and "Cancel".

12. 必要に応じて、**【選択】** をクリックして、プロセスオブジェクトを保存する場所を指定します。  
デフォルトは、**【OpenAPI 3.0 ファイルからの新規プロセスオブジェクト作成】** ウィザードを開いたときに選択された場所です。
13. 必要に応じて、プロセスオブジェクトに対し追加のタグをカンマ区切り値として指定します。

14. **【インポートの開始】** をクリックします。

アプリケーション統合では、プロセスオブジェクトのインポートが開始され、インポートステータスが表示されます。

次の図は、インポートステータスを示しています。



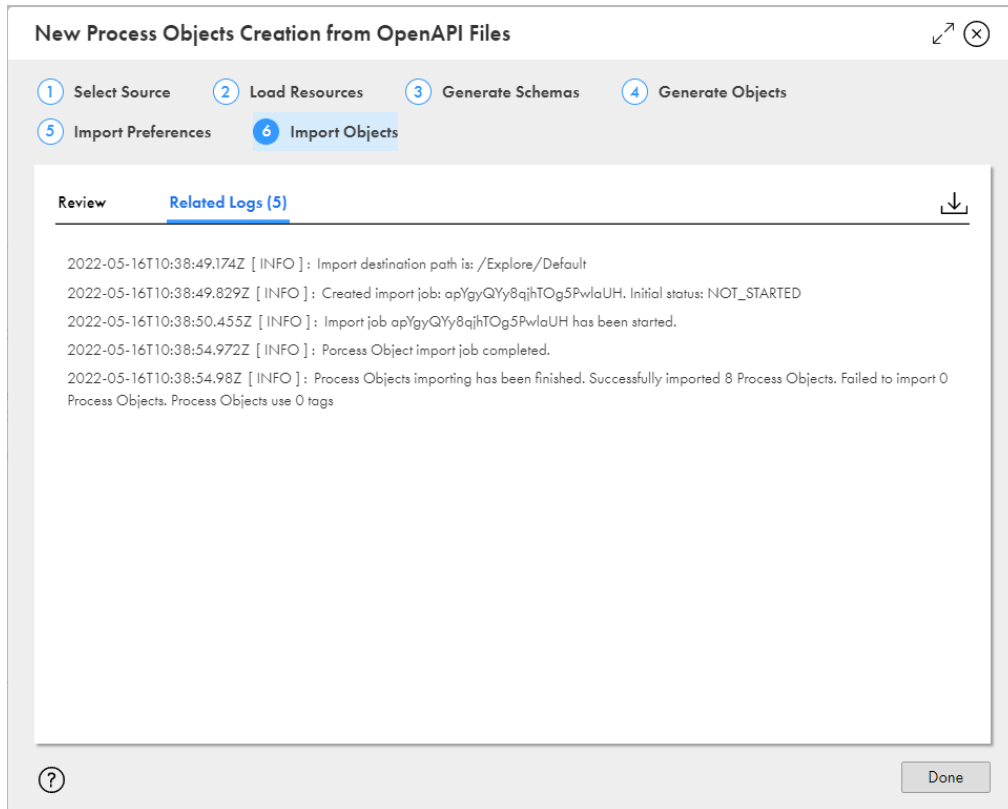
15. 以下のいずれかのアクションを実行します。

- ダイアログボックスを破棄し、プロセスオブジェクトのインポート完了時に通知を受けるには、**【破棄】** をクリックします。
- インポートエラーを表示し、修正してインポートを再試行するには、ダイアログボックスを開いたままにします。

プロセスオブジェクトのインポート後、アプリケーション統合ではインポートが成功または失敗したかどうかを示すステータスメッセージが表示されます。

インポートが失敗した場合、**【関連ログ】** タブを確認してエラーを修正し、プロセスオブジェクトのインポートを再試行します。**【ダウンロード】** アイコンをクリックして、ログ全体をダウンロードすることもできます。

次の図は、8つのプロセスオブジェクトが正常にインポートされたことを示すサンプルログを示しています。



16. **【完了】** をクリックします。
17. プロセスオブジェクト生成に指定した場所に移動し、プロセスオブジェクトを確認します。

## JSON ファイルからのプロセスオブジェクトの作成

1つ以上のJSONファイルをインポートすることで、複数の大きな階層プロセスオブジェクトを簡単に同時に作成できます。

**【JSON ファイルからの新規プロセスオブジェクト作成】** ウィザードを使用して、プロセスオブジェクトをJSONファイルから作成します。ソースを指定すると、アプリケーション統合ではソースが解析され、リソースがロードされて、XSDスキーマが生成されます。XSDスキーマに基づいて、アプリケーション統合ではプロセスオブジェクトが生成されます。プロセスオブジェクトを保存する場所などのインポート設定を指定し、必要に応じてプロセスオブジェクトに使用されるタグを指定できます。

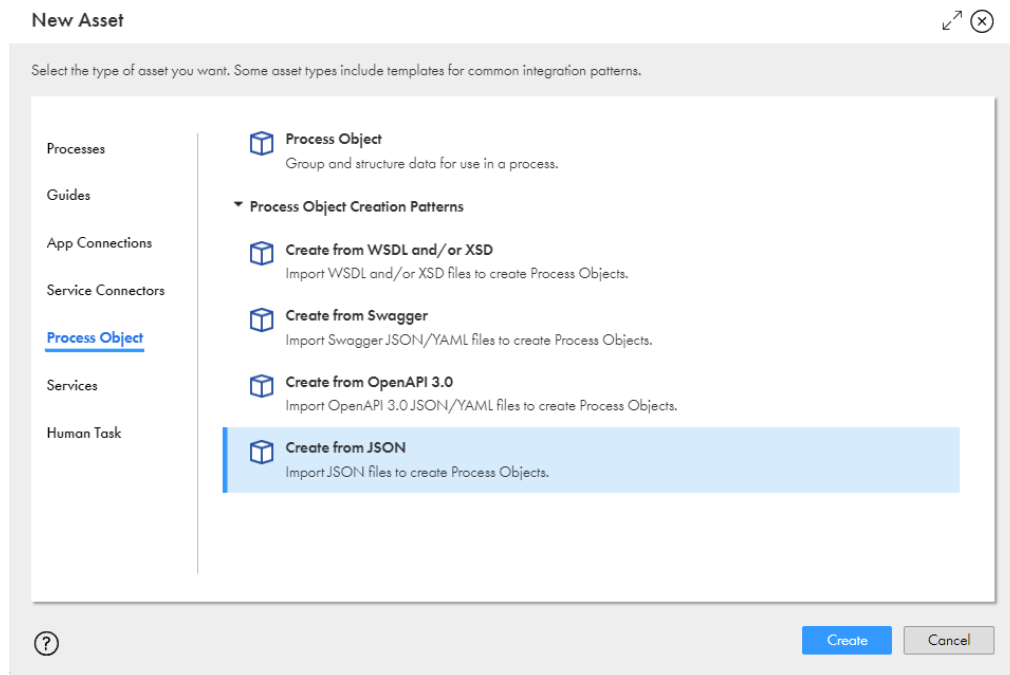
プロセスオブジェクト作成中にエラーが発生した場合、ウィザードの該当するステップの**【関連ログ】** タブにエラーが表示されます。ログを確認して対処できます。ログをダウンロードすることもできます。特定のステ

ップでログをダウンロードすると、ログにはそれより前のすべてのステップに関する累積ログ情報も表示されます。

1. アプリケーション統合で、**【新規】** をクリックします。

**【新しいアセット】** ダイアログボックスが表示されます。

次の図は、**【新しいアセット】** ダイアログボックスを示しています。

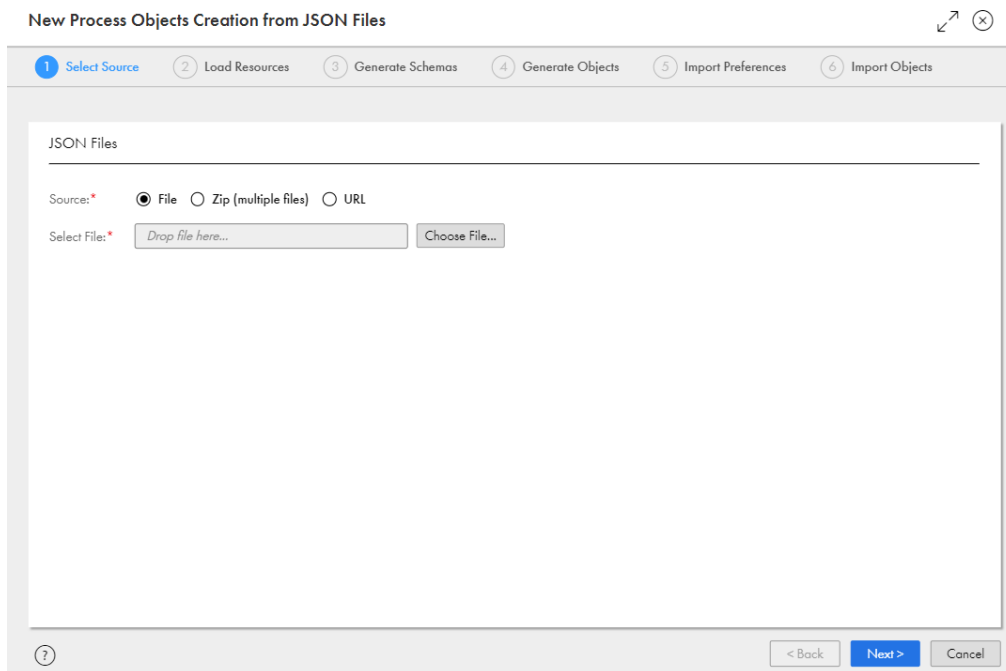


2. 左ペインの **【プロセスオブジェクト】** をクリックします。

3. 右ペインで、**【プロセスオブジェクト作成パターン】** リストを展開し、**【JSON から作成】** をクリックしてから、**【作成】** をクリックします。

**【JSON ファイルからの新規プロセスオブジェクト作成】** ダイアログボックスが表示されます。

次の図は、**【JSON ファイルからの新規プロセスオブジェクト作成】** ダイアログボックスを示しています。



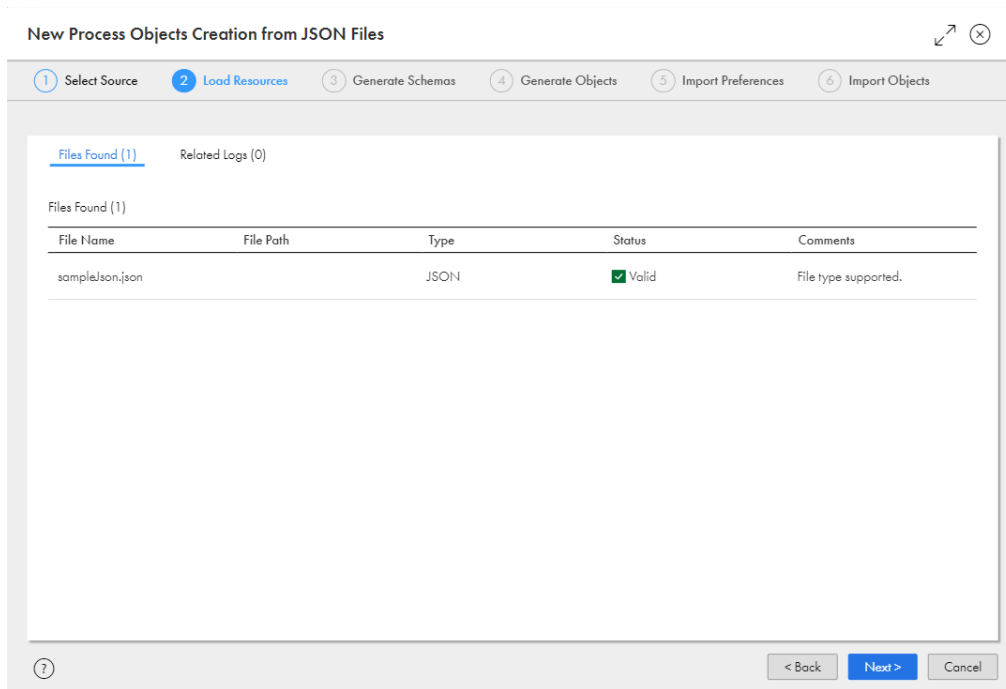
4. **【ソース】** フィールドで、次のうちいずれかの値を選択します。

オプション	説明
ファイル	<p><b>【ファイル】</b> を選択し、ローカルシステムからペイロードを持つ単一の JSON ソースファイルを指定します。HTTP リソースを指定することもできます。<b>【ファイルを選択】</b> をクリックして、JSON ソースファイルを選択します。</p> <p>単一ファイルの最大許容ファイルサイズは 5 MB です。</p> <p>デフォルトは <b>【ファイル】</b> です。</p>
Zip (複数ファイル)	<p>ローカルシステムから、プロセスによって受信または返されるペイロードのサンプルを含む、zip ファイルの形式の複数の JSON ファイルを指定するには、<b>【Zip】</b> を選択します。<b>【ファイルを選択】</b> をクリックして、zip ファイルを選択します。</p> <p>zip ファイルの最大許容ファイルサイズは 3 MB です。</p>
URL	<p>ペイロードを持つ JSON ソースファイルが含まれている URL を指定するには、<b>【URL】</b> を選択します。<b>【URL】</b> フィールドと <b>【認証の使用】</b> フィールドが表示されます。</p> <p><b>【URL】</b> フィールドに、ソース URL を入力します。</p> <p><b>【認証の使用】</b> オプションを選択して、ソース URL にアクセスするためのユーザー名とパスワードを指定します。</p>

5. **【次へ】** をクリックします。

アプリケーション統合では指定されたソースファイルまたは URL が解析され、**【リソースのロード】** タブにリソース詳細が表示されます。zip ファイルをインポートすると、**【リソースのロード】** タブに JSON リソースのリストが含まれるようになります。

次の図は、リソース詳細が表示された **【リソースのロード】** タブを示しています。

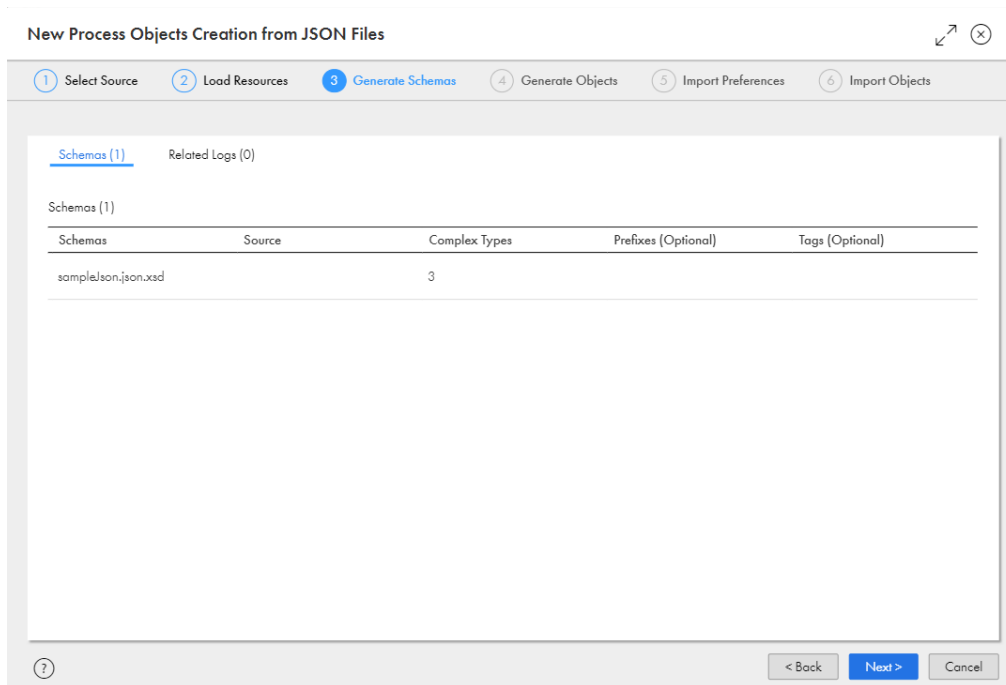


6. リソース詳細を確認します。  
以下の表に、【リソースのロード】タブに表示される列を示します。

カラム名	説明
ファイル名	リソースのファイル名。
ファイルパス	ソース内の各リソースの URL または関連する場所。
タイプ	リソースのファイルタイプ。値が【JSON】と表示されます。
ステータス	リソースのステータス。次のいずれかの値を表示します。 - <b>有効</b> 。ファイルタイプがサポートされ、ファイルコンテンツが有効であることを示します。 - <b>無効</b> 。ファイルタイプがサポートされていないか、ファイルコンテンツが無効であることを示します。
コメント	ファイルタイプがサポートされているかどうかを示します。

解析エラーがある場合、アプリケーション統合では【関連ログ】タブにエラー詳細が表示されます。エラーを修正してやり直してください。

7. 【次へ】をクリックします。  
アプリケーション統合では、作成されるプロセスオブジェクトに基づいてリソースが XSD スキーマに変換されます。【スキーマの生成】タブにスキーマ詳細が表示されます。  
次の図は、スキーマ詳細が表示された【スキーマの生成】タブを示しています。



8. スキーマ詳細を確認します。  
以下の表に、【スキーマの生成】タブに表示される列を示します。

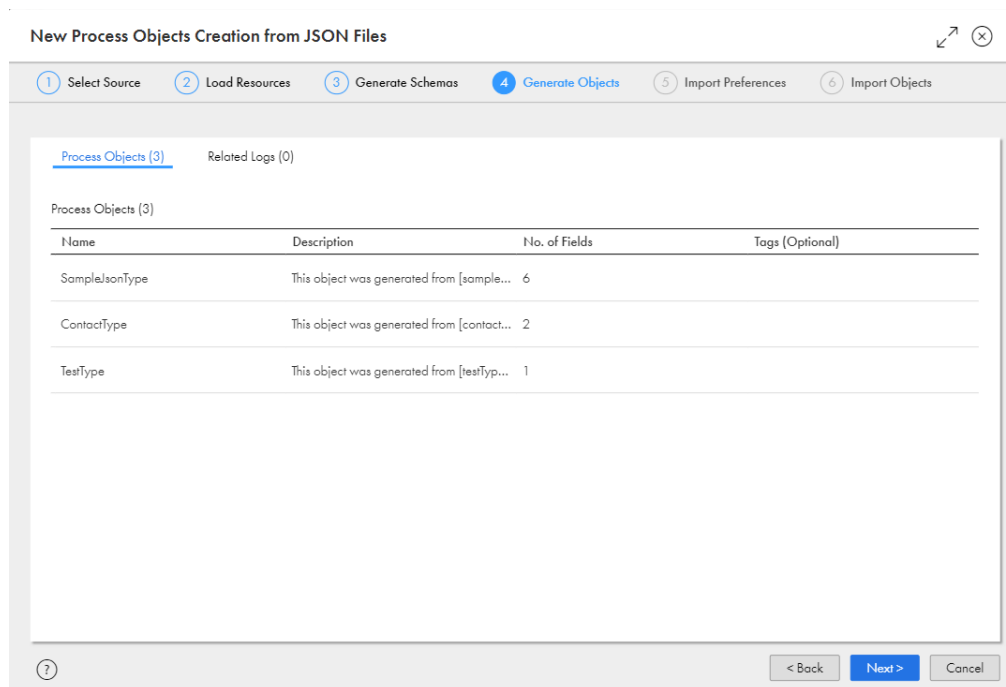
カラム名	説明
スキーマ	生成された XSD スキーマファイルの名前。
ソース	XSD スキーマファイルが生成されたソース。
複合型	XSD スキーマごとに作成される複合型の数。
プレフィックス	XSD スキーマに使用されるプレフィックス。 プレフィックスを使用して、スキーマから作成されるプロセスオブジェクトを簡単に特定できます。例えば、Salesforce スキーマから作成されるすべてのプロセスオブジェクトに対しプレフィックス Salesforce を追加するとします。 スキーマに対しプレフィックスを指定するには、スキーマが含まれている行に対し【プレフィックス】列をダブルクリックし、プレフィックスを入力します。スキーマから作成されるすべてのプロセスオブジェクトでは、このフィールドで指定したものと同一プレフィックスが使用されます。
タグ	XSD スキーマに使用されるタグ。 タグを使用して、関連するスキーマをまとめてグループ化することができます。例えば、財務リソースを含むすべてのスキーマをタグ Finance にグループ化するとします。 スキーマに対しタグを指定するには、スキーマが含まれている行に対し【タグ】列をダブルクリックし、タグを入力します。スキーマから作成されるすべてのプロセスオブジェクトでは、このフィールドで指定したものと同一プレフィックスが使用されます。

スキーマ生成エラーがある場合、アプリケーション統合では【関連ログ】タブにエラー詳細が表示されません。エラーを修正してやり直してください。

9. **【次へ】** をクリックします。

アプリケーション統合では、XSD スキーマに指定したプレフィックスとタグが適用され、プロセスオブジェクトが XSD スキーマに基づいて生成されます。**【オブジェクトの生成】** タブにプロセスオブジェクト詳細が表示されます。プロセスオブジェクト名に **Type** という用語のサフィックスが追加されます。

次の図は、プロセスオブジェクト詳細が表示された **【オブジェクトの生成】** タブを示しています。



10. プロセスオブジェクトの詳細を確認します。

以下の表に、**【オブジェクトの生成】** タブに表示される列を示します。

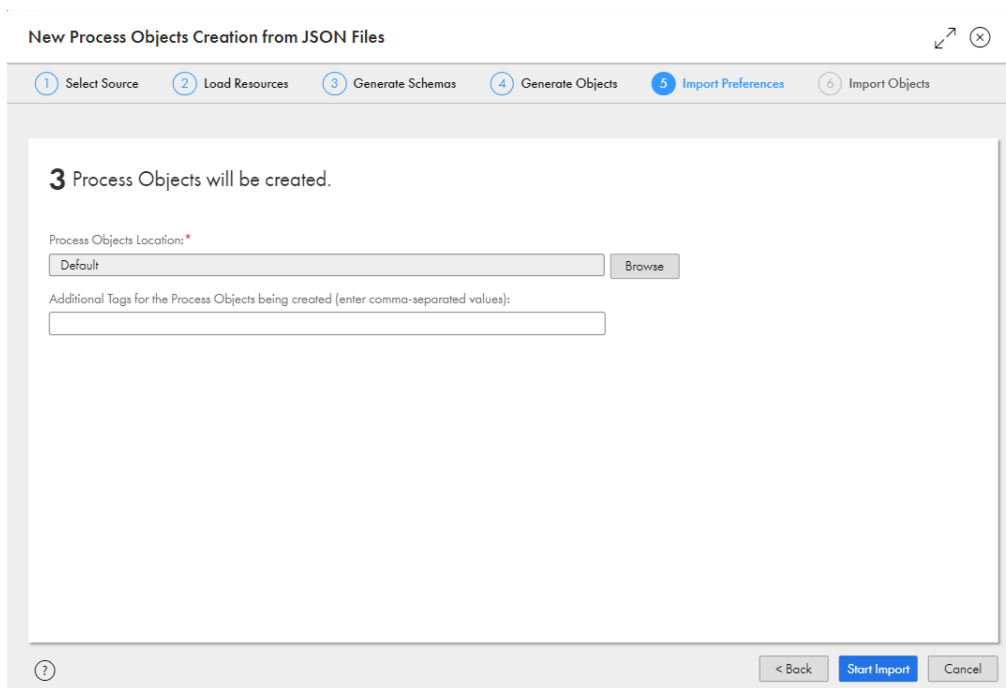
カラム名	説明
名前	スキーマから生成されたプロセスオブジェクトの名前。
説明	プロセスオブジェクトが生成されたスキーマ。説明を編集できます。
いいえフィールド関連	プロセスオブジェクトに含まれている複合型の数。
タグ	プロセスオブジェクトに使用されるタグ。デフォルトで、プロセスオブジェクトタグは XSD スキーマに指定したタグから派生します。 プロセスオブジェクトレベルで別のタグを指定する場合はタグを編集できます。 プロセスオブジェクトのタグを編集するには、プロセスオブジェクトを含む行に対し <b>【タグ】</b> 列をダブルクリックし、タグを更新します

11. **【次へ】** をクリックします。

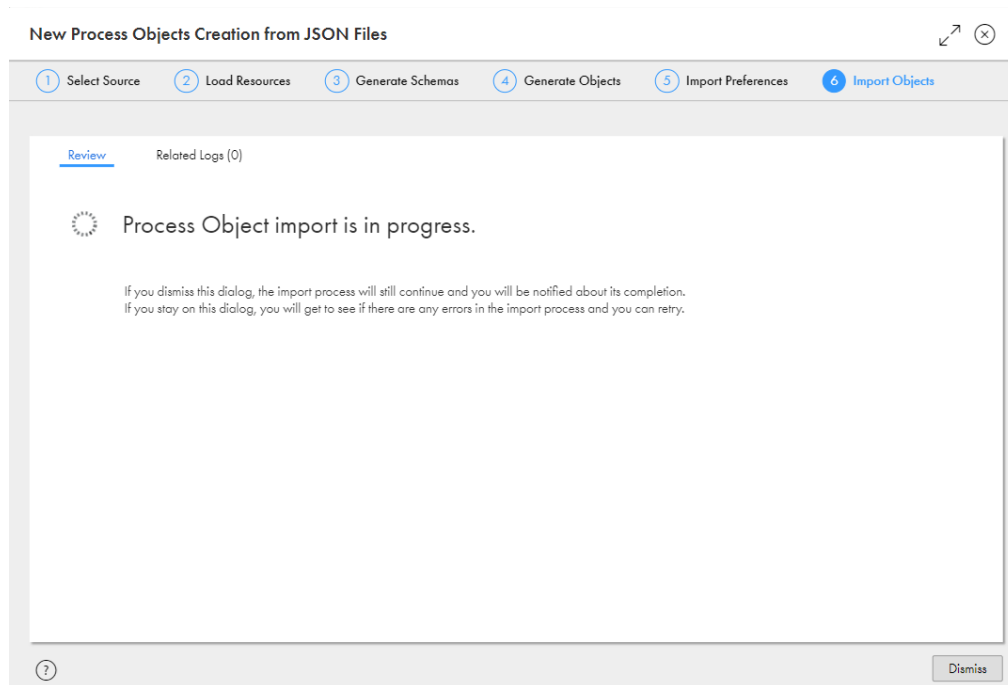
作成されるプロセスオブジェクトの数を表示する **【設定のインポート】** タブが表示されます。

次の図は、**【設定のインポート】** タブを示しています。





12. 必要に応じて、**【参照】** をクリックして、プロセスオブジェクトを保存する場所を指定します。  
デフォルトは、**【JSON ファイルからの新規プロセスオブジェクト作成】** ウィザードを開いたときに選択された場所です。
13. 必要に応じて、プロセスオブジェクトに対し追加のタグをカンマ区切り値として指定します。
14. **【インポートの開始】** をクリックします。  
アプリケーション統合では、プロセスオブジェクトのインポートが開始され、インポートステータスが表示されます。  
次の図は、インポートステータスを示しています。

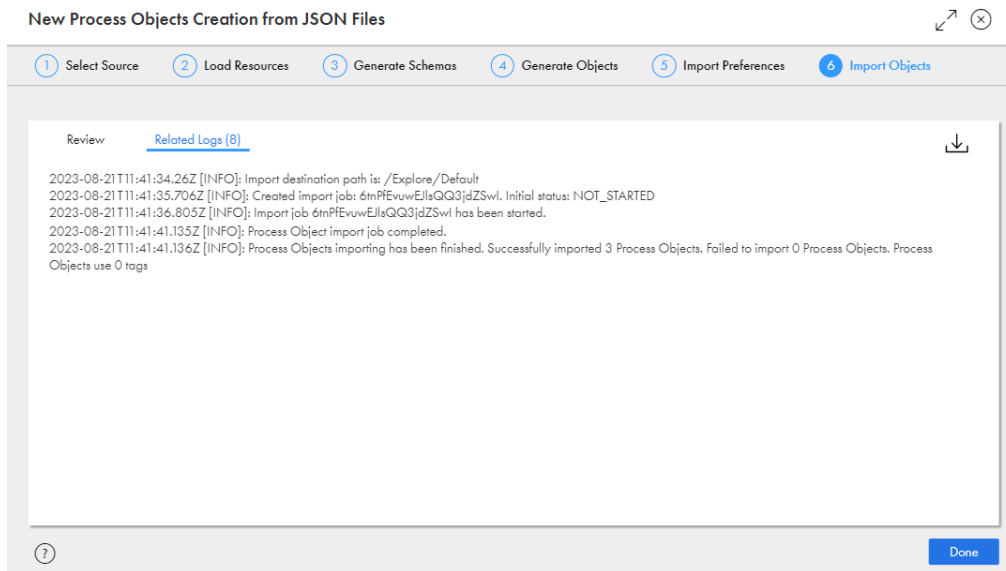


15. 以下のいずれかのアクションを実行します。
- ダイアログボックスを破棄し、プロセスオブジェクトのインポート完了時に通知を受けるには、**【破棄】** をクリックします。
  - インポートエラーを表示し、修正してインポートを再試行するには、ダイアログボックスを開いたままにします。

プロセスオブジェクトのインポート後、アプリケーション統合ではインポートが成功または失敗したかどうかを示すステータスメッセージが表示されます。

インポートが失敗した場合、**【関連ログ】** タブを確認してエラーを修正し、プロセスオブジェクトのインポートを再試行します。**【ダウンロード】** アイコンをクリックして、ログ全体をダウンロードすることもできます。

次の図は、8つのプロセスオブジェクトが正常にインポートされたことを示すサンプルログを示しています。



16. **【完了】** をクリックします。
17. プロセスオブジェクト生成に指定した場所へ移動し、プロセスオブジェクトを確認します。

## JSON ファイルのルールおよびガイドライン

JSON ファイルをインポートしてプロセスオブジェクトを作成する場合は、次のルールとガイドラインを考慮してください。

- JSON ファイルをインポートしてプロセスオブジェクトを作成すると、JSON ファイル内で定義されたオブジェクトに加えて、そのファイル名を持つルートプロセスオブジェクトも作成されます。  
sampleJson.json という名前の JSON ファイルがあり、このファイルには、次のサンプルに示すような JSON ペイロードが含まれているとします。

```

{
  "address": "India",
  "age": 25,
  "contact": {
    "telephone": 5434344985
  },
  "ID": 111,
  "name": "",
  "test": {
    "contact": {
      "mobile": 1267673455
    }
  }
}

```

sampleJson.json ファイルをインポートすると、ContactType、TestType、および SampleJsonType という名前の 3 つのプロセスオブジェクトが作成されます。

次の図は、JSON ファイルのインポート後に作成されるプロセスオブジェクトを示しています。

<input type="checkbox"/>	Name	Type	Updated On	Location	Description
<input type="checkbox"/>	ContactType	Process Object	Aug 2, 2023, 3:10 AM	Default	This object was generated from [contactType complex type] located in the [sampleJson.json.xsd] source XML schema.
<input type="checkbox"/>	SampleJsonType	Process Object	Aug 2, 2023, 3:10 AM	Default	This object was generated from [sampleJsonType complex type] located in the [sampleJson.json.xsd] source XML schema.
<input type="checkbox"/>	TestType	Process Object	Aug 2, 2023, 3:10 AM	Default	This object was generated from [testType complex type] located in the [sampleJson.json.xsd] source XML schema.

- JSON ファイル名に数字しか含まれていない場合、プロセスオブジェクト名の前に RootP0 が付加されます。例えば、ファイル名が 1234.json の場合、プロセスオブジェクト名は RootP01234 のようになります。
- 1つのファイル内に同じ名前のプロセスオブジェクトが複数ある場合、プロセスオブジェクトは1つだけ作成され、再利用されます。すべてのフィールドが、同じプロセスオブジェクトにマージされます。例えば、上記の sampleJson.json ファイルには、contact という名前のオブジェクトが2つあります。このJSON ファイルをインポートすると、telephone フィールドと mobile フィールドが ContactType プロセスオブジェクトにマージされます。
- 選択したフォルダに同じ名前のプロセスオブジェクトがすでに存在する場合は、次のメッセージがログに表示されます。  
[WARNING]: Process object <Process\_Object\_Name> imported with warning.Resolution: CONFLICT, Status: SUCCESSFUL, Status Code: MigrationSvc\_055, StatusMessage: Reuse existing object

## 必須要素と NULL 可能要素

ファイルをインポートしてプロセスオブジェクトを生成する前に、WSDL ファイル、Swagger JSON ファイルまたは YAML ファイル、および OpenAPI 3.0 JSON ファイルまたは YAML ファイルで、プロセスオブジェクトフィールドを必須および NULL 可能として定義できます。

WSDL ファイルで、フィールドを NULL 可能として定義するには、フィールド要素で nillable 属性を "true" に設定します。nillable 属性を指定しない場合、デフォルトでは、フィールドは NULL 可能ではないものとして設定されます。フィールドを必須として設定しない場合は、フィールド要素で minOccurs 属性を "0" に設定します。minOccurs 属性を指定しない場合、デフォルトでは、フィールドは必須と見なされます。

例えば、ProcessObjectReq という名前のスキーマを持つ WSDL ファイルがあるとします。このスキーマは複合型で、Name と ID という2つの要素を含んでいます。ID フィールドを NULL 可能とし、必須としない場合は、次のサンプルに示すように、nillable 属性を "true" に設定し、minOccurs 属性を "0" に設定します。

```
<xsd:complexType name="ProcessObjectReq">
  <xsd:sequence>
    <xsd:element name="Name" nillable="false" type="xsd:string"/>
    <xsd:element name="ID" minOccurs="0" nillable="true" type="xsd:integer"/>
  </xsd:sequence>
</xsd:complexType>
```

WSDL ファイルをインポートすると、次の図に示すように、WSDL ファイルで定義された NULL 可能および必須のフィールド要素がプロセスオブジェクトにプロパゲートされます。

ProcessObjectReq | Valid

Save

**Properties:**

Name: \* ProcessObjectReq

Location: \* Default

Description:

**Fields:**

Name:  Label:  Type: Text

Name	Label	Type	Required	Nullable	Delete
<a href="#">Name</a>	<a href="#">Name</a>	<a href="#">Text</a>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="button" value="X"/>
<a href="#">ID</a>	<a href="#">ID</a>	<a href="#">Number</a>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="button" value="X"/>

同様に、次のサンプルに示すように、Swagger JSON ファイルでも、必須および NULL 可能なフィールド要素を設定し、プロセスオブジェクトを作成できます。

```

"definitions" : {
  "Process1-3Request" : {
    "type" : "object",
    "required" : [ "Test" ],
    "properties" : {
      "Test" : {
        "$ref" : "#/definitions/ProcessObjectReq"
      }
    }
  },
  "ProcessObjectReq" : {
    "type" : "object",
    "properties" : {
      "Name" : {
        "type" : "string"
      },
      "ID" : {
        "type" : "integer",
        "format" : "int32"
      }
    },
    "required" : [
      "Name"
    ],
    "x-nullable" : ["ID"]
  }
}

```

次のサンプルに示すように、OpenAPI 3.0 ファイルでも、必須および NULL 可能なフィールド要素を設定し、プロセスオブジェクトを作成できます。

```

"ApiResponse" : {
  "type" : "object",
  "properties" : {
    "code" : {
      "type" : "integer",
      "format" : "int32",
      "required" : true,
      "nullable" : false
    },
  },
}

```

```
...
    },
...

```

NULL 可能フィールドがあるプロセスオブジェクトをプロセスで使用している場合に、そのプロセスの Swagger ファイルを生成しても、生成された Swagger ファイルにはプロセスオブジェクトで定義された x-nullable フィールドが含まれないことがあります。デフォルトでは、このフィールドは NULL 可能ではないものとして設定されます。そのため、NULL 可能フィールド値を保持するには、WSDL ファイルまたは OpenAPI 3.0 ファイルを使用することをお勧めします。

このプロセスオブジェクトをプロセスで使用する場合、必須および NULL 可能としてマークされたフィールドは、生成された WSDL ファイル、Swagger ファイル、および OpenAPI 3.0 ファイルで同じ制限を使用します。

JSON ファイルをインポートしてプロセスオブジェクトを作成すると、すべてのフィールドは NULL 可能としてマークされ、デフォルトでは必須ではありません。

## プロセスオブジェクトインポートのルールおよびガイドライン

WSDL ファイル、XSD ファイル、Swagger JSON ファイルまたは YAML ファイル、OpenAPI 3.0 JSON ファイルまたは YAML ファイル、および JSON ファイルからプロセスオブジェクトをインポートする場合は、次のルールとガイドラインを考慮してください。

- WSDL ドキュメント、XSD ドキュメント、Swagger ドキュメント、OpenAPI 3.0 ドキュメント、または JSON ドキュメントが正しくない場合、アプリケーション統合でプロセスオブジェクトは生成されません。
- WSDL スキーマファイルまたは XML スキーマファイルを使用する場合、アプリケーション統合ではプロセスオブジェクトが複合型から生成され、要素宣言は使用されません。
- アプリケーション統合では、列挙型が選択リストに変換されます。
- アプリケーション統合では、サポートされていない型が文字列または最も近い関連する型にマッピングされます。関連ステップの **[関連ログ]** タブに、型変換に関する警告メッセージも出力されます。
- Swagger ファイルを使用する場合、アプリケーション統合では **[定義]** セクションで使用可能な型定義からのみプロセスオブジェクトが生成されます。アプリケーション統合では、パス宣言のインラインで使用される型に対しプロセスオブジェクトは生成されません。
- 複数の型で同じ名前を共有する場合、アプリケーション統合では重複を避けるために生成されたプロセスオブジェクト名にインデックスが追加されます。
- プロセスオブジェクトをインポートする場合、同じ場所の既存プロセスオブジェクトを上書きすることはできません。アプリケーション統合では、重複するプロセスオブジェクトはインポートされず、既存のプロセスオブジェクトが代わりに使用されます。
- WSDL ソースファイルが含まれる URL から、ピリオド (.) 文字を含む名前のプロセスオブジェクトをインポートすると、アプリケーション統合ではピリオド (.) 文字は次の文字列に置換されます。

`_dot_`

例えば、プロセスオブジェクト名が `Cart.CartStatus` の場合、名前は次のように変わります。

`Cart_dot_CartStatus`

- OpenAPI 3.0 ソースファイルが含まれる URL から、ピリオド (.) 文字を含む名前のプロセスオブジェクトをインポートすると、アプリケーション統合では、ピリオド (.) 文字はアンダースコア ( \_ ) 文字に置換されます。

例えば、プロセスオブジェクト名が `Dept.DeptID` の場合、名前は次のように変わります。

`Dept_DeptID`

- 日本語の文字が含まれるファイル名を持つプロセスオブジェクトをインポートすると、アプリケーション統合はプロセスオブジェクトの作成に失敗します。

## 第 6 章

# サービスコネクタのデザイン

Process Designer を使用してすると、API がある場合、データ統合、ファイアウォールの内側、クラウド内のプログラムやサービスに API がある場合はそれら进行操作できます。例えば、Google は提供するサービスにアクセスするためのさまざまな API を提供しています。

次の 2 種類のサービスコネクタを作成できます。

### サービスコネクタ

REST または SOAP API を使用して、サードパーティサービスに接続するためのサービスコネクタを作成できます。

サービスコネクタを作成するには、次のいずれかの方法を使用します。

- 手動でサービスコネクタを作成する。
- WSDL ファイル、Swagger JSON ファイル、OpenAPI 3.0 JSON ファイルまたは YAML ファイルをインポートして、サービスコネクタを生成する。
- GitHub からサービスコネクタをダウンロードする。
- Informatica Marketplace からサービスコネクタをダウンロードする。
- Administrator からサービスコネクタをダウンロードする。

### データアクセスサービスコネクタ

Secure Agent でデータアクセスサービスコネクタを作成し、データベースから直接データにアクセスできます。

データアクセスサービスコネクタを使用して、次のデータベースに接続できます。

- IBM DB2
- Microsoft SQL Server
- MySQL
- Oracle
- PostgreSQL

データアクセスサービスコネクタは、データベースと直接対話するためのデータベースプロパティを定義し、変数を指定することで作成できます。このサービスを使用すると、指定したデータソースで SQL CRUD 操作（選択、挿入、更新、削除など）を実行し、データベーステーブル、式、結果セット、および添付ファイルのカラムに基づいてデータソースから応答を受け取る呼び出しアクティビティを作成できます。

これらのプロセスとサービスを使用するには、次の手順を実行します。

1. サービスコネクタまたはデータアクセスサービスコネクタ（Process Designer とサービスの間の相互の関係を定義するオブジェクト）を作成して、これらのコネクタがデータを送受信できるようにします。
2. プロセスがサービスにアクセスするために使用するアプリケーション接続を定義します。



3. サービスステップを、サービスを使用する場所とするプロセスに追加します。

例えば、次の図に示す手順では、Get Vehicle Detail アクションを使用して Edmunds から情報を取得していません。

Get Makes Properties

Service Type: Connection

Connection: Edmunds2

Action: Get Makes

Description: No description

Input Fields

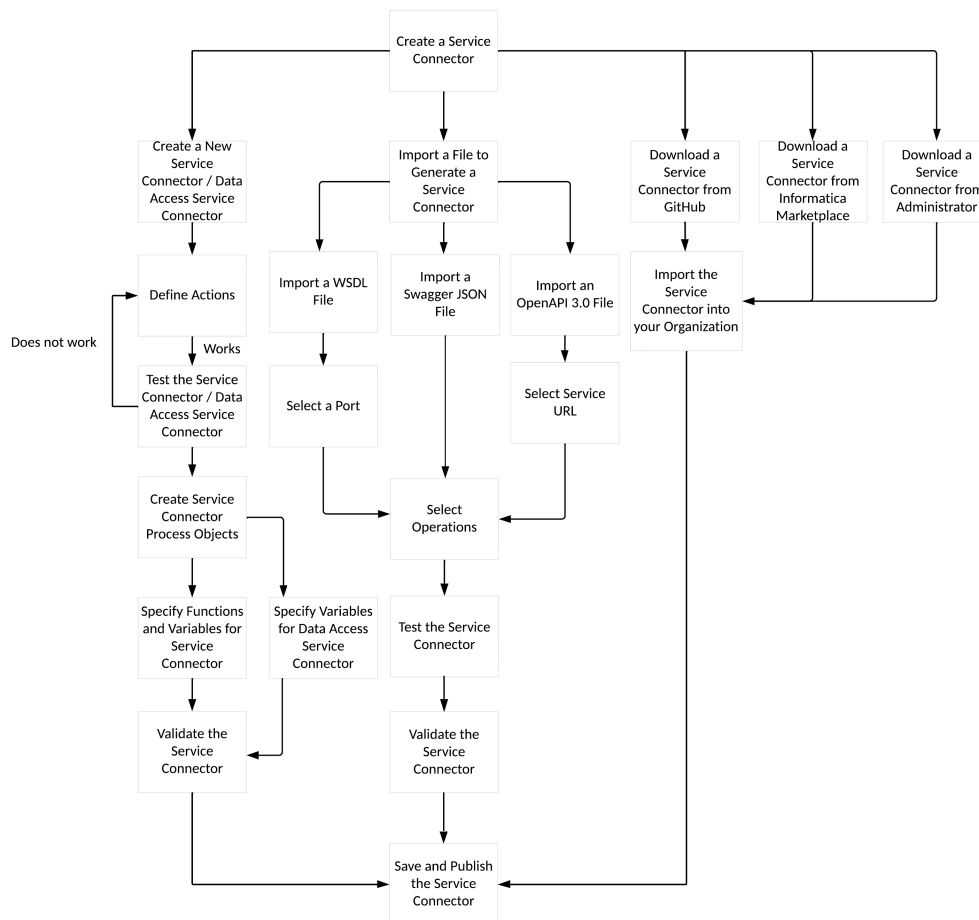
Name	Required	Type	Description
Year	<input type="checkbox"/>	Text	1990 - present

Output Fields

Name	Type	Description
MakesResult	Object ID [Edmunds2.root]	

サービスコネクタまたはデータアクセスサービスコネクタを使用して有効にしたサービスは、プロセスからアクセスできます。

次の図は、サービスコネクタの作成プロセスを示しています。

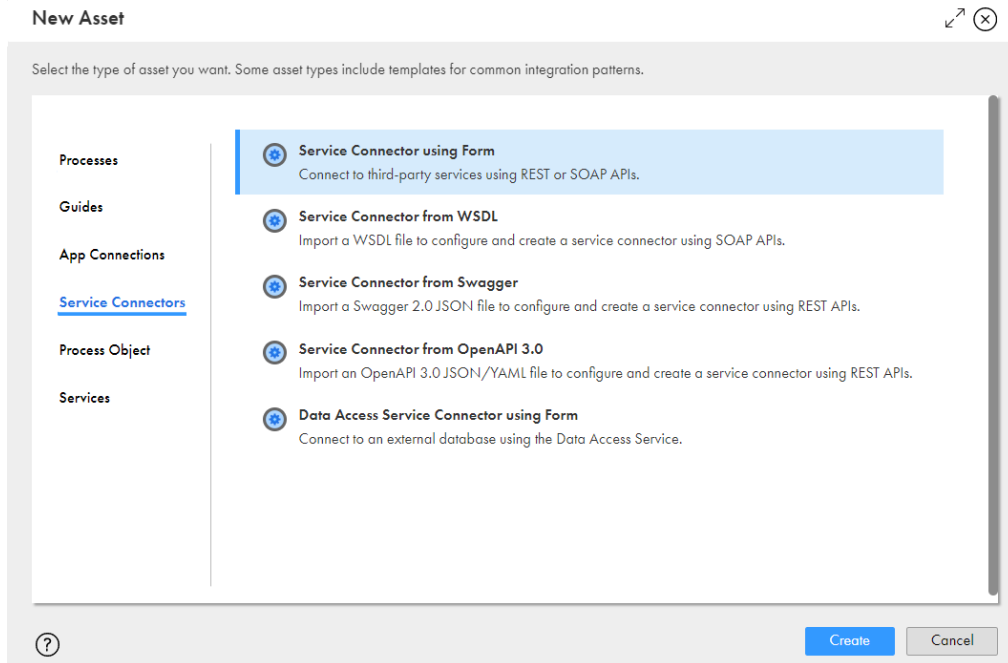


# オプション 1: 新しいサービスコネクタの作成

プロパティを定義し、関数や変数を指定すると、サービスコネクタを作成できます。また、サービスコネクタと関連付けられたアクションを定義することもできます。

サービスコネクタを作成するには、次の手順を実行します。

1. アプリケーションの統合で、**[新規]** > **[サービスコネクタ]** をクリックします。
2. **[新しいアセット]** ダイアログボックスで、**[フォームを使用したサービスコネクタ]** をクリックし、**[作成]** をクリックします。



## プロパティの定義

サービスコネクタを作成してプロパティを定義するには、次の手順を実行します。

1. アプリケーションの統合で、**[新規]** > **[サービスコネクタ]** をクリックします。
2. **[新しいアセット]** ダイアログボックスで、**[フォームを使用したサービスコネクタ]** をクリックし、**[作成]** をクリックします。
3. **[定義]** タブで、サービスコネクタの次の基本プロパティを定義します。
  - **名前:** サービスコネクタをプロセス用に使用できるようにする名前。名前の先頭の文字は英字または数字にする必要があり、名前には英数字、マルチバイト文字、スペース、アンダースコア ( \_)、およびハイフン ( -) のみを含めることができます。128 文字を超える名前は使用できません。これは必須フィールドです。
  - **場所:** サービスコネクタを保存するプロジェクトまたはフォルダを指定します。
  - **説明:** サービスコネクタの説明を入力します。
  - **エージェントのみ:** このサービスコネクタを Secure Agent で実行するためにのみ使用できるようにする必要がある場合に選択します。

4. **[+]** をクリックして、このサービスに接続するための接続プロパティを追加します。**[接続プロパティ]** セクションで、各接続プロパティについて次の詳細を指定します。
  - **名前:** 接続プロパティの名前。名前の先頭の文字は英字または数字にする必要があり、名前には英数字、スペース、アンダースコア ( \_ )、およびハイフン ( - ) のみを含めることができます。これは必須フィールドです。
  - **説明:** 接続プロパティの説明を入力します。
  - **テストに使用する値:** Process Designer が接続をテストするために使用する値を入力します。
  - **タイプ:** 接続プロパティに対して適切なデータ型を選択します。
  - **必須:** 接続プロパティを必須のプロパティとして設定するには、このチェックボックスをオンにします。
  - **暗号化:** 接続プロパティを暗号化するには、このチェックボックスをオンにします。開発者コンソールには、コネクタを保存するときにこれらのフィールドに入力した値が表示されます。コネクタを保存して閉じ、再度開くと、値は暗号化されて表示されます。

次の図に、サービスコネクタのサンプルを示します。

The screenshot shows the configuration window for a service connector named 'ConnectWithInheritance'. The 'Definition' tab is selected. The 'Name' field contains 'ConnectWithInheritance', 'Location' is 'Default', and 'Agent Only' is checked. Below is a table of connection properties:

Name	Description	Test With	Type	Required	Encrypt
APIKey *	API Key	AJO25PQ738494	string	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Name *	Name		string	<input checked="" type="checkbox"/>	<input type="checkbox"/>

5. 接続プロパティを編集するには、接続プロパティを含んだ行の任意の場所をクリックします。行のすべてのプロパティが編集可能になります。
6. **[保存]** をクリックします。

サービスコネクタを定義するアクションおよびプロセスオブジェクトの詳細については、[「アクションの定義」 \(ページ 189\)](#) および [「サービスコネクタプロセスオブジェクトの作成」 \(ページ 201\)](#) を参照してください。

## 関数と変数の指定

サービスコネクタを構成するには、バインディング、出力フィールド、その他のプロパティを定義するための変数と関数を指定する必要があります。

## ビルトイン変数

これらの変数は、XQuery 式を使用してサービスコネクタで参照できます。

変数	変数型	説明
<code>\$VariableName</code>	すべての接続プロパティ	プロパティ、入力、その他のパラメータは、この形式を使用して指定できます。
<code>\$ResponseStatusCode</code>	出力フィールドマッピング	HTTP 応答コード。
<code>\$ResponseHeaders</code>	出力フィールドマッピング	各項目が次のとおりの要素リスト内に HTTP 応答ヘッダーが格納されます。 <header name="Content-Type">text/plain</header> 以下に例を示します。 <code>\$ResponseHeaders[@name="Content-Type"]/text()</code>
<code>\$RESTResponse</code>	出力フィールドマッピング	ヘッダーとコードを含む RESTResponse XML データが格納されます。サービスコネクタのテスト結果内に表示されます。

## 出力フィールドマッピング関数

サービスコネクタでバインディングを定義するときは、式エディタで多くの関数を使用できます。

使用可能なすべての機能については、「[関数の使用](#)」(ページ 29)および「[式エディタの使用](#)」(ページ 23)を参照してください。

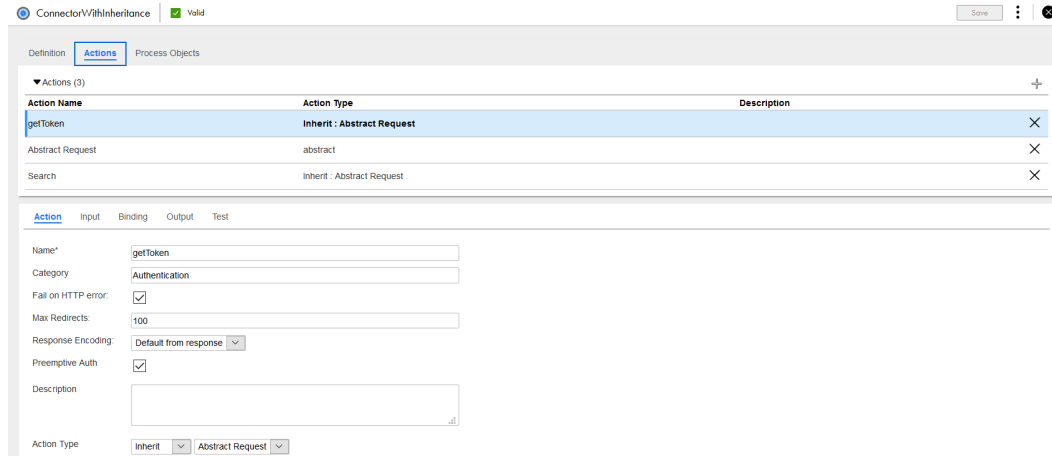
次の表に、サービスコネクタの出力フィールドのマッピングで使用できる関数を示します。

機能	構文	説明
<code>responseHeaderExists</code>	<code>svc:responseHeaderExists(\$ResponseHeaders, headerName) : boolean</code>	ヘッダーパラメータが応答内にあるかどうかを示す Boolean 値を返します。
<code>getResponseHeader</code>	<code>svc:getResponseHeader(\$ResponseHeaders, headerName, defaultValue) : string</code> <code>svc:getResponseHeader(\$ResponseHeaders, headerName) : string</code>	応答ヘッダー値を返します。ヘッダーパラメータが定義されていない場合、この関数は、オプションのデフォルト値を返します。
<code>getResponseHeaderNames</code>	<code>svc:getResponseHeaderNames(\$ResponseHeaders) : list of strings</code>	応答ヘッダー内のすべてのパラメータの名前を格納したリストを返します。

## アクションの定義

[アクション] タブで、サービスコネクタと関連付ける 1 つまたは複数のアクションを作成して記述できます。

次の図は、[アクション] タブを示しています。



編集する行を選択し、次の情報を [アクションの詳細] タブに入力します。

[+] をクリックし、新しい行を追加します。

- **アクション名:** このアクションの参照時にサービスコネクタおよび接続のリストに表示される名前を入力します。名前の先頭の文字はアルファベットまたは数字にする必要があります。また、名前に使用可能な文字は英数字、スペース、およびアンダースコア ( \_ ) のみです。これは必須フィールドです。
- **カテゴリ:** 多数のサービスコネクタがあり、それぞれに複数のアクションがある場合、ユーザーがプロセス内で移動しやすくなるように、カテゴリを作成できます。
- **HTTP エラー時に終了:** このアクションを使用するサーバーへの要求の送信時にエラーが発生した場合に、Process Designer が認識するように設定するには、このオプションを選択します。詳細については、[「HTTP エラーのチェック」 \(ページ 199\)](#) を参照してください。
- **最大リダイレクト数:** アクションが処理できるリダイレクトの最大数を指定します。アクションがエンドポイントにアクセスするときに、エンドポイントはアクションを別のエンドポイントにリダイレクトすることがあります。リダイレクト数を制御するための値を入力します。デフォルトの **最大リダイレクト数** の値は 100 です。  
リダイレクトを無効にするには **0** を入力します。  
例えば、**3** を入力した場合、サービスコネクタアクションは最大数である 3 つのリダイレクトを作成し、エンドポイントから 3 つのリダイレクト応答を受け取ります。エンドポイントが 4 番目のリダイレクト応答を要求した場合、エラーメッセージが表示されます。
- **プリエンティブ認証:** 認証の詳細 (ユーザー資格情報) がエンドポイントへの要求と一緒に送信されます。エンドポイントに認証が必要なことがわかっている場合、**プリエンティブ認証** を選択します。**プリエンティブ認証** を選択せず、エンドポイントに認証が必要な場合、次のイベントが発生します。
  1. サービスコネクタは要求をエンドポイントに送信します。
  2. エンドポイントは認証を要求します。
  3. サービスコネクタはユーザー資格情報をエンドポイントに送信します。**プリエンティブ認証** を選択し、エンドポイントに認証が必要な場合、余分な要求が発生することを避けられます。

- **アクションタイプ:** 各アクションに対し、次のいずれかのアクションタイプを選択できます。詳細については、「[共有サービスアクション](#)」(ページ 190)を参照してください。
  - **[全般]** アクションは、特定のサービスコネクタと一緒に使用する目的でのみパブリッシュされます。
  - **[要約]** アクションはパブリッシュされませんが、他のアクションと共有できます。つまり、他のアクションのテンプレートとして再利用できます。
  - **[継承]** アクションは、要約アクションからプロパティを継承します。
- **説明:** このアクションの説明または付記を入力します。

各アクションについて、**[入力]**、**[バインディング]**、**[出力]**、**[テスト結果]** の各タブに、追加の詳細を指定します。

## 共有サービスアクション

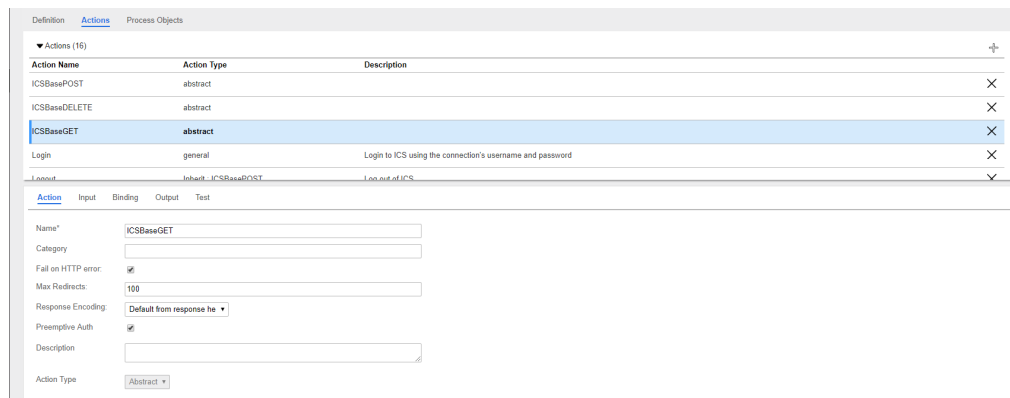
共有サービスアクションを使用すると、共通の入力、バインディング、出力を定義し、親「抽象」アクションから「継承する」その他のアクションでこれらの定義を再利用できます。

例えば、次の同じ要素を共有する複数のアクションを設けられます。

- 入力パラメータ session-id
- HTTP ヘッダー Content-Type など
- バインディングメソッド (GET/POST) および URL
- 認証資格情報
- 出力フィールド

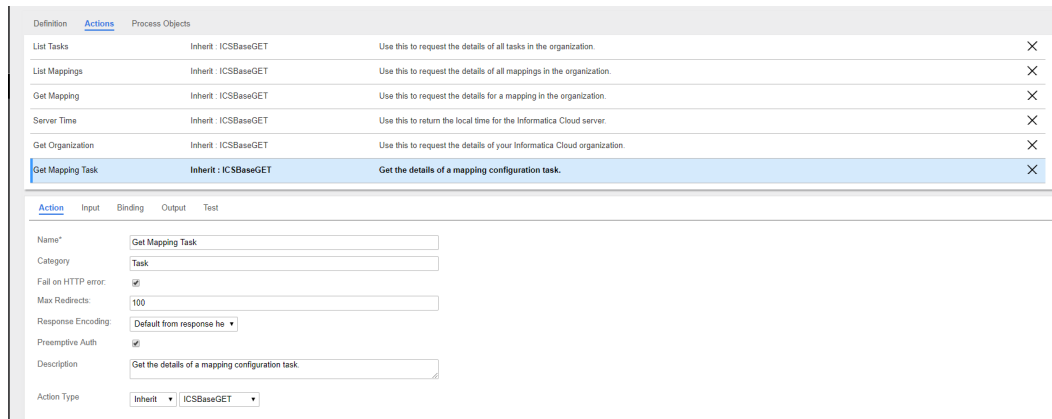
この場合、次のように実行できます。

1. すべての共通定義を含む抽象型アクションを定義します。



2. これらの共通定義を使用する複数の継承タイプのアクションを作成し、特定のパラメータを拡張または追加します。
3. 例えば、継承アクションを次のようにできます。
  - 新しい入力パラメータ (\$messageId など) を定義します。
  - バインディング URL を拡張し、「/{messageId}」を使用して新しいパラメータをベースに追加します。

以下に例を示します。



## 使用方法に関する注意

共有サービスアクションは次のとおりです。

- IPD とガイド内で設計時には使用できません。
- 入力、バインディング、出力は単一の親（抽象）アクションから継承します。
- 抽象アクションが他の抽象アクションから継承することは、許可されません。
- 各サービスコネクタが複数の抽象アクションを持つことは許可されます。

継承アクションタイプを指定すると、多くのフィールドが無効になります。

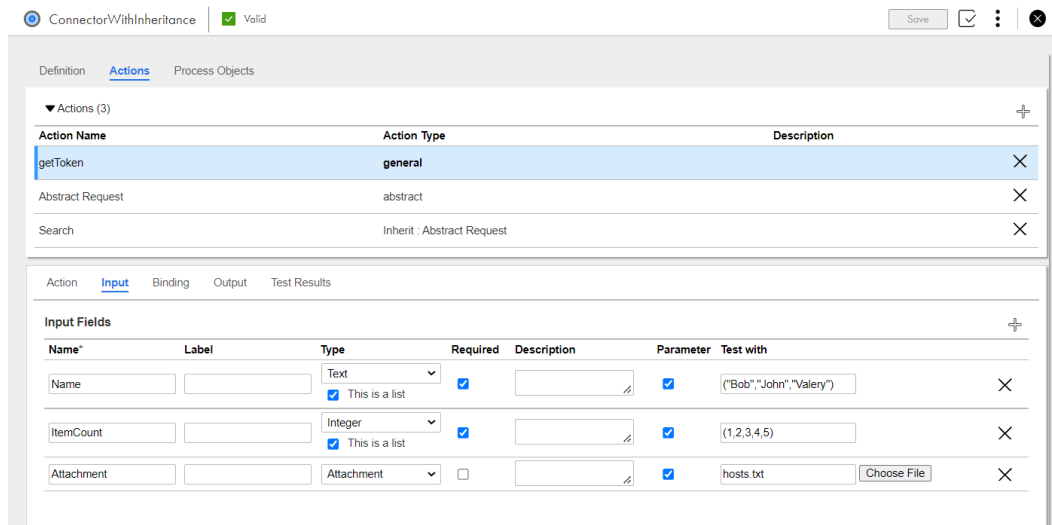
ただし、特定のバインディングやその他のオプションの場合、次を選択できます。

- **継承:** 抽象アクションからのすべての入力パラメータをそのまま使用します。これがデフォルトです。
- **除外:** 設計時または実行時にアクションに使用できないように、パラメータを除外します。
- **上書き:** 継承したパラメータを上書きするパラメータを指定します。

## [入力] タブ

[入力] タブを使用して、データの送信先サービスに固有の入力データ項目を定義します。

以下に例を示します。



サービスによっては、データ項目がない場合や、データ項目が多数ある場合があります。

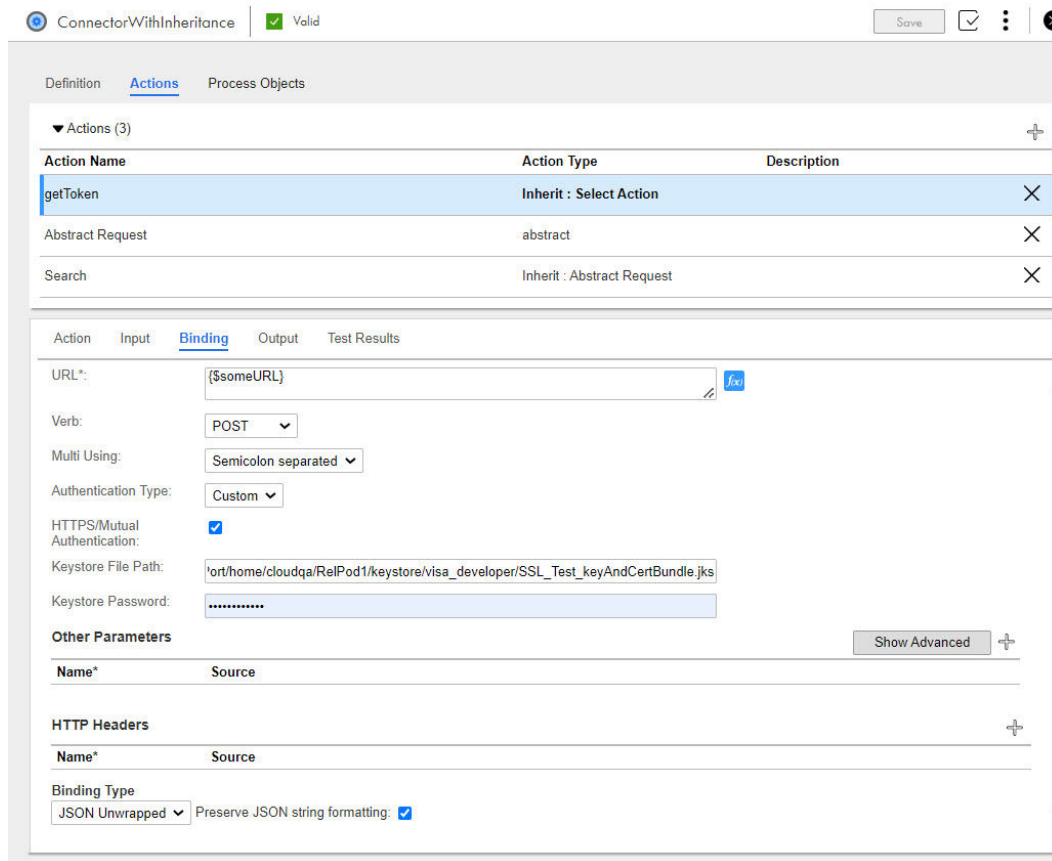
各項目に対し、次のプロパティを入力します。

- **名前:** 入力データ項目の名前。名前は、文字、数字、特殊文字.@\${}\_ を含めることができ、先頭にそれらのいずれかを使用する必要があります。名前にはスペースを含めることもできます。これは必須フィールドです。
  - **ラベル:** プロセス内に表示する項目の名前。指定しない場合、[名前] が表示されます。
  - **タイプ:** リストから項目のデータ型を選択します。[参照] または [オブジェクトリスト] 型を選択した場合、プロセスオブジェクトも選択できます。  
入力フィールドにテキスト、整数、数値、チェックボックス、日付、日時、時刻などの単純型のリストを設定することもできます。初期値には、次の例に示すように、単純型のリストに基づいた一連の値を使用します。
    - テキストのリスト: ("Bob", "John", "Valery")
    - 整数のリスト: (1,2,3)
    - 数値のリスト: (22.5, -23.654, -45.876500)
    - チェックボックスのリスト: (true, false, true)
    - 日付のリスト: ("2022-01-03", "2021-12-11", "2020-03-04")
    - 日時のリスト: ("2022-01-03T11:23:01Z", "2021-12-11T23:11:55", "2020-03-04T22:10:45")
    - 時刻のリスト: ("12:30:00", "22:45:23", "05:01:02")型を単純型のリストとして設定するには、必要に応じて [テキスト]、[整数]、[数値]、[チェックボックス]、[日付]、[日時]、または [時刻] を選択し、[これはリストです] チェックボックスを選択します。
  - **必須:** このパラメータに値を設定する必要がある場合に選択します。
  - **説明:** パラメータの説明を入力します。
  - **パラメータ:** 選択した場合、サービスコネクタはこのパラメータをサービスに渡します。パラメータが別のパラメータの作成時にのみ使用されるようにし、サービスに渡される必要がない場合は、このフィールドの選択を解除します。例えば、サービスは日付を RFC 1123 形式で受け取りますが、そのサービスを呼び出すプロセスが、Process Designer の通常の形式である XSD の日付を渡すようにします。このオプションは、XML を除くすべてのデータ型に対して有効です。
  - **テストに使用する値:** Process Designer がアクションをテストするために使用する値を入力します。入力が添付ファイルの場合、サンプルの添付ファイルをアップロードすることもできます。
- [+] アイコンを使用し、新しい項目を追加します。[X] をクリックして現在の行を削除します。

## サービスコネクタバインディング

サービスコネクタ内の各アクションに対し、バインディングオプションを使用することで、サービスと通信するために必要なインタフェースとパラメータを指定できます。





## バインディングプロパティ

次のオプションに値を指定します。

- **URL:** REST サービスの URL を入力します。
- **動詞:** サービスに対して次のデータの送信方法を選択します。
  - **GET、HEAD、OPTIONS、TRACE:** Process Designer はクエリパラメータを自動的に生成します。
  - **POST、PUT、PATCH、および DELETE:** Process Designer は **[バインディングタイプ]** フィールドを追加します。そのフィールドで、値を **[JSON]**、**[JSON ラップなし]**、**[フォーム]**、**[URL]**、または **[カスタム]** として選択できます。**[カスタム]** を選択した場合、**[本文]** フィールドが表示され、サービスに送信する要求を入力できます。デフォルトでは、**POST、PUT、および PATCH** 動詞では **[JSON]**、**DELETE** 動詞では **[なし]** が選択されています。入力内容については、API のドキュメントを参照してください。
- **マルチ使用:** このオプションによって、パラメータに複数の値を指定するときのクエリ文字列パラメータの生成方法が決まります。次のいずれかを選択します。
  - **セミコロン区切り:** セミコロン区切りの複数選択リストの場合、Process Designer は入力パラメータ値を単一の文字列として追加します (例: `?param=a;b;c`)。これがデフォルトです。
  - **カンマ区切り:** カンマ区切りの複数選択リストの場合、Process Designer は入力パラメータ値を単一の文字列として追加します。例: `?param=a,b,c`。
  - **括弧の追加:** セミコロン区切りの複数選択リストの場合、Process Designer はセミコロン区切りの値を同じ名前の複数のクエリパラメータにマッピングし、パラメータ名に `[]` を追加します。例: `colors[]=red&colors[]=blue&name=JW`。

- **数字の追加:** セミコロン区切りの複数選択リストの場合、Process Designer はセミコロン区切りの値を同じ名前の複数のクエリパラメータにマッピングし、フィールド名の末尾に数字を追加します。つまり、1..N です。例: `colors1=red&colors2=blue&name=JW`。
- **認証タイプ:** サービスに必要な認証情報を [基本] または [カスタム] の 2 つのオプションを使用して定義します。選択すると次のようになります。
  - [基本] の場合、要求された場合はユーザー名とパスワードを入力します。  
テキストまたは XQuery 式をユーザー名フィールドおよびパスワードフィールドに入力します。XQuery 式を入力した場合、パスワード値をサービスコネクタ内にハードコードする代わりに、接続を介して渡されます。
  - [パスワードの表示] を選択すると、パスワードの値が画面に表示されます。[パスワードの表示] が選択された状態でサービスコネクタをエクスポートすると、パスワードもエクスポートされます。
  - [カスタム] の場合、サービス要件に基づいて、セキュリティパラメータを追加します。例えば、署名付きデータを HTTP ヘッダーを使用して送信する必要がある場合があります。
- **HTTPS/相互認証:** アクションで相互認証にカスタムクライアント証明書を使用する必要がある場合は、このチェックボックスを選択します。アクションごとに 1 つのカスタムクライアント証明書を設定できます。選択した場合、[キーストアファイルパス] フィールドと [キーストアのパスワード] フィールドが表示されます。キーストアファイルを Secure Agent マシンに配置する必要があります。
 

**注:** Secure Agent で実行されるサービスコネクタのカスタムクライアント証明書を設定できます。サービスコネクタのカスタムキーストアファイルを設定し、さらに Process Server プロパティの `javax.net.ssl.keyStore` プロパティを変更するときは、サービスコネクタで設定されたカスタムキーストアファイルが優先されます。サービスコネクタの認証に失敗した場合、`javax.net.ssl.keyStore` プロパティで設定したクライアント証明書は無視され、要求は失敗します。
- **キーストアファイルパス:** Secure Agent マシンの Java KeyStore (JKS) ファイルのパスとファイル名を入力します。
- **キーストアのパスワード:** キーストアファイルを開くためのパスワードを入力します。

### その他のパラメータ

このセクションは、サービスに必要な入力パラメータを追加するために使用します。この値は、リテラルまたは式として指定できます。使用しているサービスからのドキュメントを参照してください。

行を追加し、複数のパラメータを入力するには、[+] 記号をクリックします。行を削除するには、[X] をクリックします。

各行に次の値を入力します。

- **名前:** 呼び出されるサービス内に指定されたとおりのパラメータの名前を指定します。
- **コンテンツ:** パラメータに指定する値がある場合はその値を指定します。

バインディングのためのオプションのパラメータを選択するには、[詳細の表示] をクリックします。

- **署名に使用する値:** 計算される値は、最初に SHA1 を使用して署名され、署名の結果がこの値になります。
- **署名に使用する値:** SHA1 または SHA256 を使用してフィールドを署名する場合に選択します。
- **エンコードに使用する値:** Base64 および Hex64Upper としてフィールドをエンコードする場合に選択します。
- **一時:** 選択した場合、GET または POST の使用時に Process Designer はパラメータを生成しません。詳細については、「[\[入力\] タブ](#)」(ページ 191)に記述されたパラメータに関する情報を参照してください。
- **Attachment Base64:** <inputName>と関連付けられた添付の Base64 エンコードを指定します。
- **次からの添付ファイル名:** <inputName>と関連付けられた添付のフィールド名を指定します。

フィールドを非表示にするには、[詳細の非表示] をクリックします。

## HTTP ヘッダー

サービスに 1 つまたは複数の HTTP ヘッダーが必要な場合、このセクションの各ヘッダーに、[名前] と [コンテンツ] (パラメータの値) を入力します。

SOAP ベースのサービス接続を定義するには、「SOAPAction」ヘッダーと「Content-Type」ヘッダーの両方を追加する必要があります。SOAPAction の値は空にする、GET を使用して動的に設定する、WSDL で定義することができます。

サービスの応答ヘッダーに Content-Type が設定されていない場合、サービスコネクタはコンテンツタイプをペイロードから推論しようとします。例えば、応答が{...}または[...]のいずれかで始まるか終わっている場合、その応答は JSON として解析されます。応答のペイロードが山括弧 (<) で始まっている場合、その応答は XML として扱われます。応答のペイロードが不具合のある JSON または XML であるために解析できない場合、通常の文字列として処理されます。

応答 Content-Type ヘッダーの最初のパートに **application** という用語、後続のパートに **json** という用語が含まれている場合、アプリケーション統合では応答ペイロードタイプが JSON として推測されます。同様に、応答 Content-Type ヘッダーの最初のパートに **text** という用語、後続のパートに **xml** という用語が含まれている場合、アプリケーション統合ではペイロードタイプが XML として推測されます。

**注:** [名前] フィールドでの入力時に、Process Designer はよく使用される値を表示します。これには次のような値があります。

- Accept
- Accept-Charset
- Accept-Encoding
- Accept-Language
- Content-Type
- SOAPAction

一部のサービスには、MD5 署名を要求するものがあります。MD5 署名は、サービスコネクタアクションの **[HTTP ヘッダー]** セクションで生成できます。

MD5 署名を生成するには、次の手順を実行します。

- HTTP ヘッダーの名前を入力します。例えば、「**Signature**」と入力します。
- **[ソース]** > **[式エディタ]** に移動します。
- **[タイプ]** の下で、**[XQuery]** を選択します。
- **[フィールドの挿入]** の下で、**[PAYLOAD\_DIGEST\_MD5]** を選択します。

サービスコネクタをテストまたは実行するときに、使用される変数に生成済み要求ペイロードの MD5 チェックサムが格納されます。

## コンテンツタイプと文字セット

要求ペイロードに使用する文字セットは、ここに示すように判断できます。

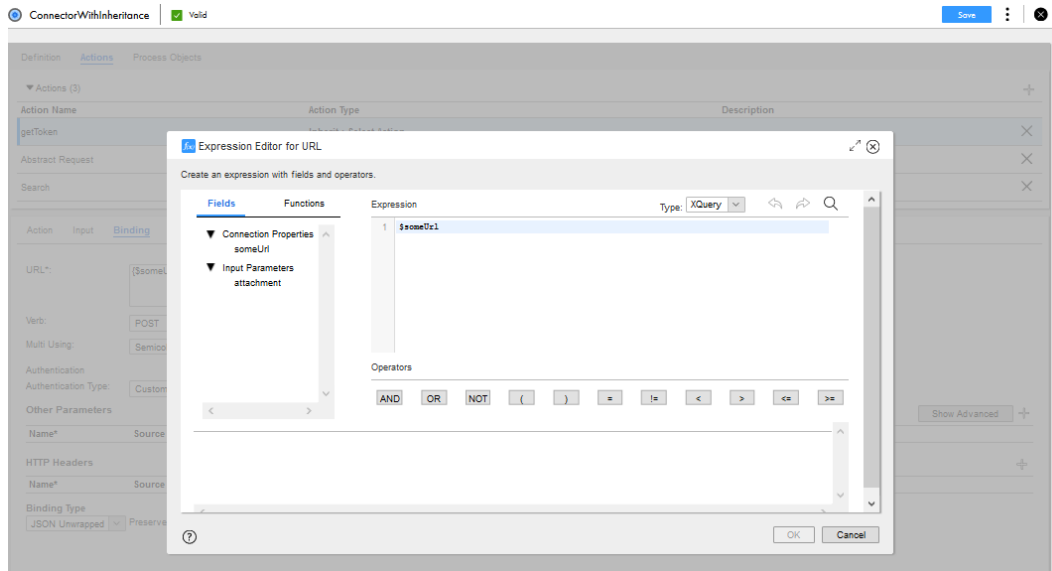
- 文字セットの値を Content-Type ヘッダー内に指定します。この場合、ヘッダーによって、要求ペイロードの文字セットが決まります。この場合、文字セットは、要求の形式が添付タイプでない限り、ヘッダーに指定される値に基づきます。添付タイプである場合、multipart/form-data 要求の各部分は個別に扱われます。Content-Type を設定し、サービスコネクタがデフォルトの文字セットを自動的に付加するようにする場合、次に示すように、指定した Content-Type の後にセミコロンを追加します。  
Content-Type=application/json;

- Content-Type ヘッダーには文字セットを指定せず、この値が要求形式に基づいて自動的に設定されるようにしてください。コンテンツが正しくデコードされることを保証するため、サービスコネクタはデフォルトの文字セット (UTF-8) を Content-Type に付加します。例えば、この値は次のように設定されます。  
Content-Type=application/json;charset=UTF-8

## サービスコネクタプロパティの式エディタ

バインディングパラメータを定義するには、[URL]、[その他のパラメータ]、[HTTP ヘッダー] の各フィールドに対して使用できる式エディタを使用できます。

式エディタを開くには、編集するフィールドの横の **[f (x)]** をクリックします。式エディタを開くと、図に示すように、使用可能な関数と、サービスコネクタに対して定義されたフィールドのリストにアクセスできます。



式に選択された型に基づいて、式エディタで構文検証が適用されます。この場合、post\_url フィールドは、XQuery 変数として検証されます。

詳細については、「[式エディタの使用](#)」 (ページ 23) を参照してください。

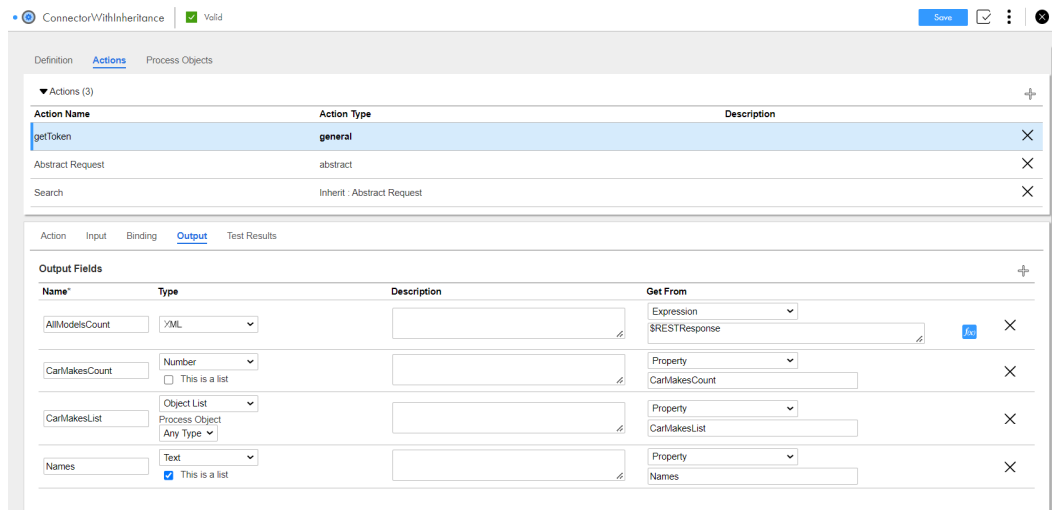
## [出力] タブ

**[出力]** タブを使用して、サービスコネクタがサービスから返されるデータを解析し、そのデータを変数に配置する方法を定義します。通常、サービスはデータを XML 形式で返します。サービスがデータを JSON 形式で返す場合、サービスコネクタはそれを XML 形式に変換します。XML は、次の 2 種類の方法で出力フィールドにマッピングできます。

- XML タグ名または JSON プロパティに基づいて、データを直接プロパティにマッピングします。
- 式を使用して、要素を抽出します。

**注:** JDOM や Woodstox などの XML 解析ツールを使用して XML 応答を解析してから、値を抽出する (プレーン文字列検索を実行して名前空間とタグを照合するのではなく) ことをお勧めします。名前空間のプレフィックスが一定ではないため、プレーン文字列を照合するとエラーが発生する可能性があります。

次の図は、**[出力]** タブを示しています。



各出力データ項目について、次の要素を指定します。

- 名前:** 返された値が配置される変数の名前。名前の先頭の文字はアルファベットまたは数字にする必要があります。また、名前に使用可能な文字は英数字、スペース、およびアンダースコア ( \_ ) のみです。これは必須フィールドです。
- タイプ:** 変数に書き込まれる値のデータ型。タイプが [オブジェクトリスト] または [参照] である場合、Process Designer はプロセスオブジェクトのリストを表示し、[プロセスオブジェクト] タブ内で定義されたオブジェクトのいずれかを選択できるようにします。出力フィールドにテキスト、整数、数値、チェックボックス、日付、日時、時刻などの単純型のリストを設定することもできます。型を単純型のリストとして設定するには、必要に応じて [テキスト]、[整数]、[数値]、[チェックボックス]、[日付]、[日時]、または [時刻] を選択し、[これはリストです] チェックボックスを選択します。
- 説明:** 変数を説明するテキスト。
- 取得元:** 次のいずれかのオプションを選択します。
  - プロパティ:** 変数内に配置される名前付きの値（サービスから返される XML 内で使用される名前）を入力できるようにします。
 

**注:** 一意のフィールドでは [プロパティ] のみを選択する必要があります。重複したフィールドがある場合は、[式] を選択してから、XPath 式を指定します。詳細については、[「単純型のリストを使用する際のルールとガイドライン」 \(ページ 198\)](#) を参照してください。
  - 式:** サービスから返される XML を解析するための式を記述できるようにします。上の図では、AllModelsCount 変数の 1 つの値を返すために、\$RESTResponse 出力フィールドが使用されています。CarMakesList に使用された式は、リストを返します。[f(x)] をクリックして、式を入力するための式エディタを開きます。
  - HTTP 応答ステータスコード:** HTTP 応答ステータスコードを確認できるようにします。
  - HTTP 応答ヘッダー:** フィールドに割り当てられる応答ヘッダーの部分を入力できるようにします。詳細については以下を参照してください。
  - 応答全体:** フィールドに応答ペイロードのコンテンツ全体を割り当てられるようにします。
  - 簡易 XML:** データを再配置することで、プロセスオブジェクトが使用できるようにします。詳細については、[「簡易 XML」 \(ページ 199\)](#) を参照してください。
  - 添付ファイルとしてのすべての応答:** 応答全体を添付ファイルとして処理します。
 

**注:** [添付ファイルとしてのすべての応答] オプションでは、下位互換性が維持されます。以前に単一の添付ファイルを持つサービスコネクタを作成した場合、そのファイルはそのまま機能するため、手動で変

更する必要はありません。ただし、添付ファイルを手動で分割する必要がないよう、サービスコネクタを再設計して、マルチパート応答が複数の添付ファイルとして処理されるようにすることをお勧めします。また、プロセス内ではなく、サービスコネクタ自体内で添付ファイルに処理ロジックを使用することもできます。そのため、複数のプロセス内でサービスコネクタを使用でき、複数のプロセスで処理ロジックを設計する必要がありません。出力添付ファイルの処理に使用できる関数を使用することもできます。

- **添付ファイル:** 複数の添付ファイルを処理して、添付ファイルの全リストを選択した変数（ペイロードとして使用されるパートを除く）に渡します。

HTTP エラーの処理方法の詳細については、「[「HTTP エラーのチェック」 \(ページ 199\)](#)」を参照してください。

## 単純型のリストを使用する際のルールとガイドライン

サービスコネクタで単純型のリストを使用する場合は、次のルールとガイドラインを考慮してください。

- ペイロード応答に一意のフィールドが含まれる場合にのみ、出力フィールドの【プロパティ】オプションを使用する必要があります。重複したフィールドがある場合、または単純型のリストを含むフィールドがある場合、XML ドキュメント内に同じフィールド名を持つ値が 1 つ以上ある可能性があります。その場合、【式】を選択し、**XQuery** を使用して正しいフィールドを取得する必要があります。例えば、次のサンプルに示すように、ペイロード応答の【プロパティ】オプションを選択するとします。

```
<users>
<user1>
  <ID>1</ID>
  <ID>2</ID>
  <ID>3</ID>
</user1>
<user2>
  <ID>4</ID>
  <ID>5</ID>
  <ID>6</ID>
</user2>
</users>
```

単純型のフィールドを使用し、出力を ID として設定する場合、アプリケーション統合は ID のインスタンスを識別し、値が 4 の ID といった単一の出力値を返します。ただし、単純型のリストを持つフィールドを使用する場合、アプリケーション統合は出力インスタンスを見つけます。次に、アプリケーション統合は、要素内の同じレベルからすべてのインスタンスを返します。この例では、ユーザー 1 で値が 1 の ID が識別された場合、アプリケーション統合はユーザー 1 内の ID 2 と 3 も返します。

この問題を回避するには、ID などの重複フィールドがあって、ユーザー 2 の 2 番目の ID を使用する場合は、次のサンプルに示すように **XQuery** を使用し、正しいフィールドを返すようにします。

```
./user2/ID[2]/text()
```

単純型の変数のリストの値のシーケンスを使用するには、次のシナリオを検討してください。

文字列のリストを含むテキスト出力変数がある場合、カンマ区切りの文字列を構築するには、次のサンプルに示すように XQuery を使用します。

```
{string-join($TextsOut , ",")}
```

日時値のリストがあり、日付のリストから 2 番目の値にアクセスする場合は、次のサンプルに示すように XQuery を使用します。

```
{$DateTimesOut[2]}
```

整数値のリストにあるすべての整数を反復処理する場合は、次のサンプルに示すように XQuery を使用します。

```
{for $j in $Integers
return $j}
```

- 単純型のリストを含む入力フィールドを使用して JSON ペイロードを生成すると、出力は JSON 配列の形式になります。JSON 応答を解析すると、フィールドも単純型のリストに変換されます。

例えば、生成された JSON ペイロードは次のサンプルのように表示されます。

```
{
  "userNames": ["Bob", "John", "Valery"]
}
```

XML ペイロードを生成すると、リクエストは次のサンプルのように表示されます。

```
<root>
  <userNames>Bob</userNames>
  <userNames>John</userNames>
  <userNames>Valery</userNames>
</root>
```

## 簡易 XML

Process Designer では、標準 XML でプロセスの設計者にデータを提示するために必要とされる、名前空間、属性、兄弟、子が混在する複雑な構成の多くは、必要ありません。場合によって、[取得元] に XQuery オプションを使用する必要がありますが、多くのアプリケーションでこれは必要ありません。

簡易 XML は、XML 内のデータを再配置し、プロセスオブジェクトで使用できるようにします。この再配置では、属性を子として扱い、名前空間を削除し、同じ名前の兄弟を連続させ、CDATA セクションにテキストを配置します。

[取得元] のオプションに [簡易 XML] を選択すると、Process Designer にテキストフィールドが表示されます。このフィールドに何も入力しないと、Process Designer は受け取ったすべての XML を処理します。要素の名前を入力すると、Process Designer はこの要素で単純化を開始し、この要素とその内部にネストされた要素のみが処理されます。

ピリオド (.) を入力すると、空のフィールドとして扱われます (これがデフォルトです)。

## HTTP エラーのチェック

サーバーに要求を送信し、サーバーがこの要求を完了できない場合、エラーステータスコードが返されます。サービスコネクタが HTTP エラー (HTTP 応答内の 4xx ステータスコードと 5xx ステータスコード) を処理できる 2 種類の方法があります。

1. **[アクション]** タブで定義できる各アクションについて、下の図に示すように、[HTTP エラー時に終了] を選択できます。デフォルトでは、このオプションが有効になっています。HTTP 要求が HTTP エラーを返すと、サービスコネクタは終了します。
2. **[出力]** タブで、各アクションに対して HTTP エラーの処理方法を指定します。このオプションを実装するには、[HTTP エラー時に終了] を無効にします。

### HTTP エラー時に終了

[HTTP エラー時に終了] を有効にすると、HTTP エラーが返された場合、次のようになります。

- プロセスが失敗すると、プロセスの詳細をアプリケーション統合コンソールで表示できます。
- プロセスにダイアログが表示され、受信した HTTP エラーを示すメッセージテキストがエンドユーザーに表示されます。
- サービスコネクタは、プロセスがフォールトをスローするのと同じ形式で、フォールトをスローします。

```
<sf: faultResponse>
  <sf:code>CODE</sf:code>
  <sf:reason>REASON</sf:reason>
  <sf:details>DETAILS</sf:details>
</sf: faultResponse>
```

HTTP エラーコードは、HTTP\_NNN の形式に従います。例えば「HTTP\_404」です。理由文字列に HTTP ステータスメッセージが含まれます。

**注:** 内部エラーまたはシステムエラーは、SERVICE\_CONNECTOR\_ERROR と次の理由文字列のいずれかを返します。

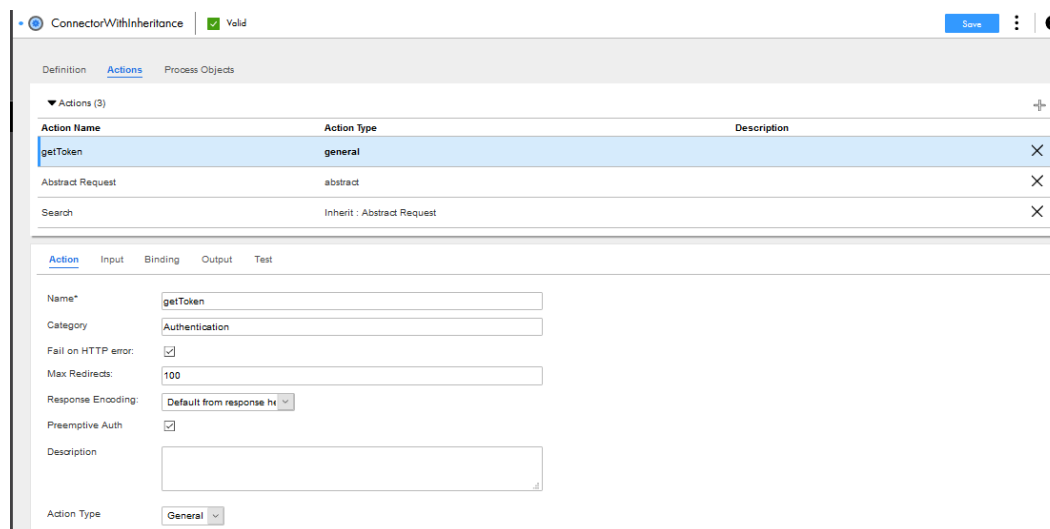
- CATALOG\_ERROR
- OTHER\_PARAMETERS\_ERROR
- AUTHENTICATION\_ERROR
- CUSTOM\_HEADERS\_ERROR
- BINDING\_URL\_ERROR
- ALTER\_REQUEST\_AUTHENTICATION\_ERROR
- PROCESS\_RESPONSE\_AUTHENTICATION\_ERROR
- OUTPUT\_PARAMETERS\_ERROR

[HTTP エラー時に終了] を無効にすると、HTTP エラーが返された場合、次のようになります。

- プロセスはフォールトにならず、サービスコネクタは 4xx ステータスコードまたは 5xx ステータスコードをプロセスに渡すため、これに応じてユーザーがプロセス内のエラーを処理できます。
- プロセスはフォールトになり、エラーメッセージが表示されます。

ターゲットサービスが、ユーザーに頻繁に表示される可能性があるエラーコードを使用する場合、このオプションを有効にして、プロセスユーザーにわかりやすい情報が提供されるようにします。

次の図に、有効化された [HTTP エラー時に終了] オプションを示します。



## HTTP エラーの処理

プロセスの HTTP エラーを処理するには、まず **[HTTP エラー時に終了]** が無効であることを確認します。

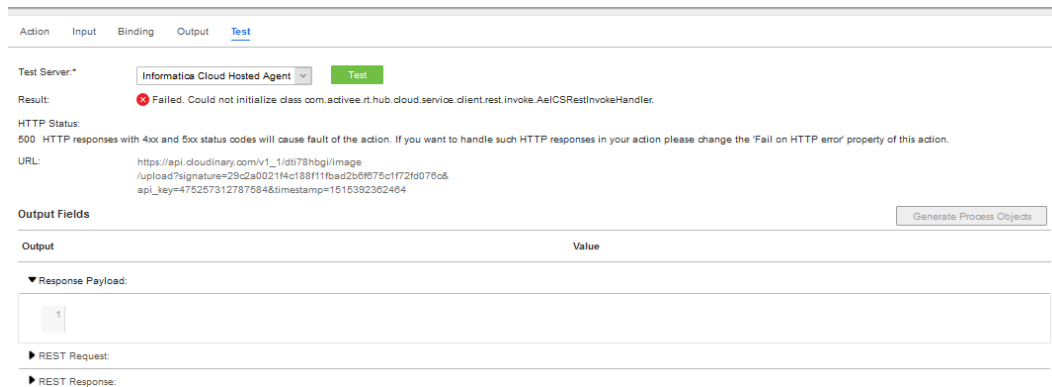
次に、プロセス内の HTTP 応答ステータスコードを確認する必要があります。

これを行うには、「HTTP 応答ステータスコード」を **[出力]** タブの変数として定義します。また、`$ResponseStatusCode` を式フィールド内で使用して、変数を指定することもできます（以下の使用可能な機能を参照）。この方法では、ステータスコードをサービスコネクタの出力の一部として渡し、プロセス内で応答を処理できます。

これらの変数は、**[出力]** タブ内で作成するすべての変数と同様に、**[テスト]** をクリックすると表示されます。

**[HTTP エラー時に終了]** を選択し、HTTP エラーステータスコードが返される場合、出力にメッセージが含まれます。





## 式を使用する HTTP 応答ヘッダー情報

次のいずれかを使用して、応答ヘッダーから詳細を取得できます。

```
$ResponseHeaders[@name="Content-Type"]/text();
```

この定数には、HTTP 応答ヘッダーが格納されます。

```
fn:getResponseHeader($ResponseHeaders, header_name[, default_value]);
```

応答ヘッダー値を返します。ヘッダーパラメータが定義されていない場合、この関数は、オプションのデフォルト値を返します。

```
fn:responseHeaderExists($ResponseHeaders, header_name);
```

ヘッダーパラメータが応答内にあるかどうかを示す Boolean 値を返します。

```
fn:getResponseHeaderNames($ResponseHeaders);
```

応答ヘッダー内のすべてパラメータの名前を格納したリストを返します。

## サービスコネクタプロセスオブジェクトの作成

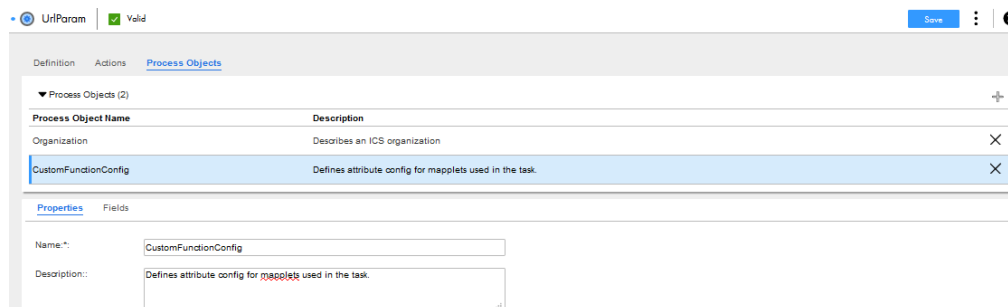
**【プロセスオブジェクト】** タブでは、サービスコネクタの 1 つ以上のプロセスオブジェクトを定義して、データをグループ化し、構造化オブジェクトを作成できます。プロセスオブジェクトがサービスコネクタ内で定義されている場合、そのサービスコネクタを使用するプロセスから使用できます。例えば、サービスが名前、住所、電話番号などの人口統計情報を返す場合、この情報を格納する単一の人口統計的プロセスオブジェクトを作成できます。

**【アクション/出力】** タブを使用して、サービスによって返された 1 つ以上のデータ要素と各プロセスオブジェクトを関連付けられます。

サービスコネクタプロセスオブジェクトを作成する手順

1. サービスコネクタを作成するか、既存のコネクタを開きます。
2. **【プロセスオブジェクト】** タブをクリックします。

3. **[+]** 記号をクリックし、新しいプロセスオブジェクトを追加するか、既存のプロセスオブジェクトをリストから選択します。各プロセスオブジェクトは、タブ内に行項目として表示されます。



4. **[プロパティ]** タブで、各プロセスオブジェクトに対し、次の詳細を指定します。
- **名前。** プロセスオブジェクトを識別するプロセスオブジェクト名を入力します。この名前は、プロセスオブジェクトを選択できるリストに表示されます。名前の先頭の文字は英字にする必要があります、名前には英数字、アンダースコア ( \_ )、およびハイフン ( - ) のみを含めることができます。これは必須フィールドです。
  - **説明。** オプションで説明を入力できます。
5. **[フィールド]** タブで、各プロセスオブジェクトに対し、次の詳細を指定します。
- **名前。** プロセスオブジェクト内の各フィールドの名前を入力します。
  - **タイプ。** 組み込みデータ型のいずれかを使用して、変数に書き込まれる値のデータ型を選択します。[オブジェクトリスト] 型の場合、プロセスオブジェクトも選択します。
  - **必要に応じて、テキスト、日付、日時、時刻、数値、整数などのデータ型を選択した後に、行を展開します。データ型ごとに次の値を入力できます。**
    - テキストデータ型の文字長とテキスト形式
    - 日付データ型の日付範囲
    - 日時データ型の日時範囲
    - 時刻データ型の時刻範囲
    - 数値データ型の最小値と最大値
    - 整数データ型の最小値と最大値
  - **必須。** このチェックボックスを選択すると、フィールドが必須フィールドとして設定されます。必須フィールドは、要求ペイロードに含まれている必要があります。値を含めることも、空にすることも、NULL にすることもできます。
  - **NULL 可能。** フィールドが NULL 値を受け入れないようにする場合は、このチェックボックスをクリアします。デフォルトでは、**[NULL 可能]** チェックボックスが選択されています。

プロセスでのプロセスオブジェクトの使用の詳細については、[「プロセスでプロセスオブジェクトを使用するためのルールおよびガイドライン」](#) (ページ 147) を参照してください。

## サービスコネクタのテスト

サービスコネクタエディタを使用するときは、**[テスト]** をクリックしてサービスに要求を送信し、応答データを **[テスト]** タブに表示します。

- **プロセスオブジェクトの生成:** クリックすると、サービスコネクタ用に定義したプロセスオブジェクトが表示されます。
- **結果:** テストのステータスを表示します。
- **HTTP ステータス:** HTTP ステータスコードを表示します。

- **URL:** テストデータの送信元 URL を表示します。ここに表示される URL に、クエリパラメータが追加されます。
- **出力および値:** **[出力]** タブで定義された変数に割り当てられたサービスから返されたデータを表示します。XSLT を使用している場合は、ペイロードからデータを抽出するために使用されます。
- **応答ペイロード:** サービスから Process Designer に送信されたペイロード（サービスから送信された完全な応答ではありません）を表示します。
- **REST 要求:** Process Designer からサービスに送信されたデータを表示します。
- **REST 応答:** サービスから Process Designer に返されたすべてのデータを表示します（応答ペイロードは、返されたデータの一部にすぎない）。

ここでは、ペイロード、要求、応答データの違いを確認できます（図示することのみを目的としており、実際のテスト結果には、要求または応答データのみが表示されます）。

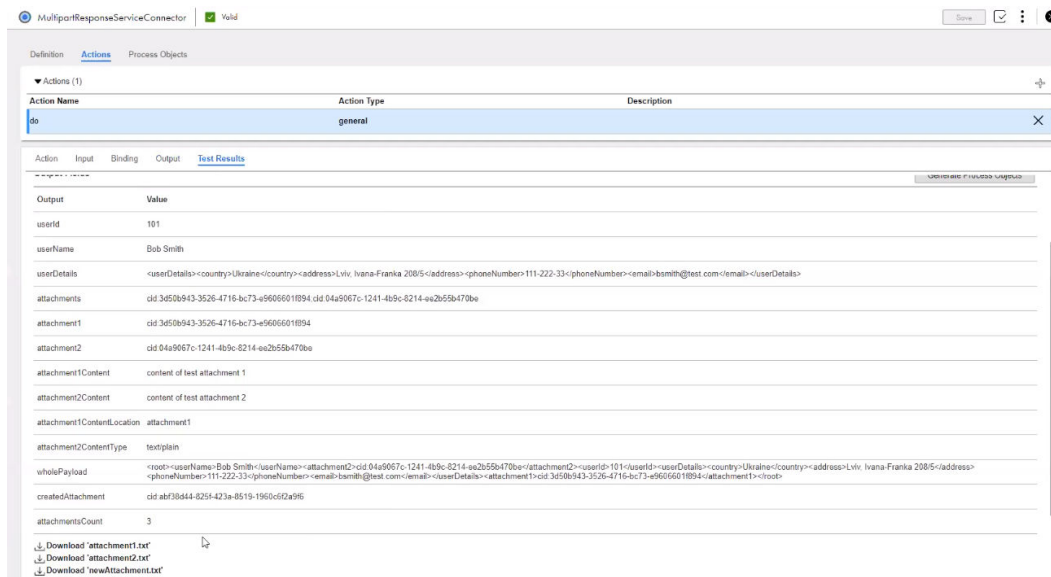
The screenshot shows the 'Test' tab of an action named 'uploadImage'. The test server is 'Informatica Cloud Hosted Agent' and the result is 'Successful'. The URL is 'https://api.cloudinary.com/v1\_1/dt78hbg/image/upload?signature=f552e01e945012120206da9014c087489208db72&api\_key=475257312787584&timestamp=1516400473551'. The REST response is as follows:

```

1 <?xml:namespace prefix="xsi" href="http://schemas.xmlsoap.org/XMLSchema.xsd" />
2 <?xml:namespace prefix="xsi" href="http://schemas.xmlsoap.org/REST/2007/12/01/aeREST.xsd" />
3 <?xml:namespace prefix="xsi" href="http://schemas.xmlsoap.org/REST/2007/12/01/aeREST.xsd" />
4 <?xml:namespace prefix="xsi" href="http://schemas.xmlsoap.org/REST/2007/12/01/aeREST.xsd" />
5 <rest:response xmlns:rest="http://schemas.xmlsoap.org/REST/2007/12/01/aeREST.xsd"
6   status="400" />
7 <rest:headers>
8   <rest:header name="Status" value="400 Bad Request" />
9   <rest:header name="X-DS-Compatible" value="IE-Edge,chrome=1" />
10  <rest:header name="Cache-Control" value="no-cache" />
11  <rest:header name="Server" value="cloudinary" />
12  <rest:header name="X-Request-Id" value="e111ee02420dbb" />
13  <rest:header name="X-Clid-Error" value="Missing required parameter - file" />
14  <rest:header name="Connection" value="keep-alive" />
15  <rest:header name="X-XSS-Protection" value="1;mode=block" />
16  <rest:header name="Content-Length" value="57" />
17  <rest:header name="Date" value="Mon, 03 Jan 2016 08:14:19 GMT" />
18  <rest:header name="Content-Type" value="application/json;charset=utf-8" />
19 </rest:headers>
20 <rest:payload contentType="application/json">{"error":{"message":"Missing required parameter - file"}}</rest:payload>
21 </?xml:namespace>

```

応答に 1 つ以上の添付ファイルが含まれている場合は、それらの添付ファイルをダウンロードすることもできます。次の図は、すべての添付ファイルをダウンロードできるマルチパート応答を示しています。



## プロセスオブジェクトの生成

通常プロセスオブジェクトは、特定の要素セットに結び付けられます。場合によっては、多数の同一のオブジェクトを持ち、それぞれが1つの要素セットに適用されることがあります。例えば、返されるデータを定義するときに、返されるデータのサブセットを定義することのみを目的とするプロセスオブジェクトを作成します。これらのオブジェクトは再利用不可能であり、単一のプロセスオブジェクトフィールドのみが使用できます。オブジェクト名は、フィールドの名前でもあります。

再利用可能なオブジェクトを作成することもできます。例えば、NetSuiteのrefType要素には、nameとinternalIDという2つのフィールドがあります。これら2つのフィールドをそれぞれが含む多数のオブジェクトを作成するのではなく、1つの再利用可能なプロセスオブジェクトを作成できます。

1. **【プロセスオブジェクトの生成】** をクリックして、サービスコネクタ用に定義したプロセスオブジェクトを表示します。
2. 再利用可能にするプロセスオブジェクトを選択し、**【次へ】** をクリックします。Process Designer に、生成されたプロセスオブジェクトの一覧が表示されます。
3. **【完了】** をクリックします。

## サービスコネクタのタイミング

サービスコネクタのタブで、HTTP 応答解析時間、HTTP 実行時間、およびリダイレクト数を参照できます。

タブの **【REST 応答】** セクションで、次の HTTP ヘッダーを参照できます。

### HTTP 応答解析時間

**【バインディング】** タブに入力した URL からの応答を解析するために、サービスコネクタが費やす時間（ミリ秒単位）。小さい要求の HTTP 応答解析時間は、通常ゼロです。大きい要求で、特に添付ファイルのある要求の場合、HTTP 応答解析時間は長くなります。

HTTP 応答解析時間は、HTTP ヘッダーの X-AE-HTTP-RESPONSE-PARSING-TIME-IN-MILLIS を参照してください。

### リダイレクト数

**【バインディング】** タブに入力した URL によって行われるリダイレクトの数（ある場合）。

例えば、サービス要求が、別のサービスにリダイレクトするサービスにリダイレクトする場合、リダイレクト数は2になります。サービス要求がリダイレクトなしで行われる場合、リダイレクト数はゼロになります。

リダイレクト数は、HTTP ヘッダーの X-AE-REDIRECTION-COUNT を参照してください。

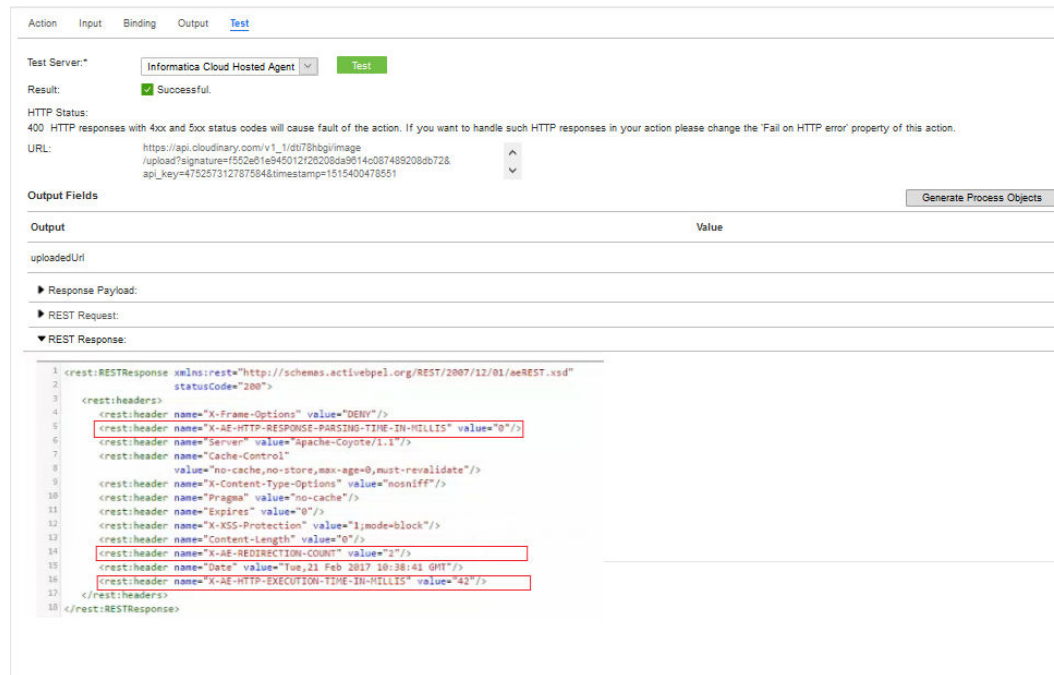
**注:** GET HTTP 要求が別の GET HTTP 要求にリダイレクトする場合、リダイレクト数はゼロのままです。これは制限事項の 1 つです。

## HTTP 実行時間

**[バインディング]** タブに入力した URL の応答時間（ミリ秒単位）。HTTP 実行時間には、応答の解析時間や、サービスコネクタが他のタスクを実行するために使用した時間は含まれません。

HTTP 実行時間は、HTTP ヘッダーの X-AE-HTTP-EXECUTION-TIME-IN-MILLIS を参照してください。

次の図は、X-AE-HTTP-RESPONSE-PARSING-TIME-IN-MILLIS、X-AE-REDIRECTION-COUNT、および X-AE-HTTP-EXECUTION-TIME-IN-MILLIS の各 HTTP ヘッダーを示します。



The screenshot shows a REST client interface with the following details:

- Test Server:** Informatica Cloud Hosted Agent
- Result:** Successful
- HTTP Status:** 400 HTTP responses with 4xx and 5xx status codes will cause fault of the action. If you want to handle such HTTP responses in your action please change the 'Fail on HTTP error' property of this action.
- URL:** https://api.cloudinary.com/v1\_1/dv78hbgj/image/upload?signature=f552e51e945012720208da9614a087499208db728&api\_key=475257312787584&timestamp=1515400478551
- Output Fields:** A table with columns 'Output' and 'Value'. The 'uploadedUri' field is present.
- REST Response:** A XML snippet showing headers:

```
1 <rest:RESTResponse xmlns:rest="http://schemas.activebpel.org/REST/2007/12/01/aeREST.xsd"
2   statusCode="200">
3   <rest:headers>
4     <rest:header name="X-Frame-Options" value="DENY"/>
5     <rest:header name="X-AE-HTTP-RESPONSE-PARSING-TIME-IN-MILLIS" value="0"/>
6     <rest:header name="Server" value="Apache-Coyote/1.1"/>
7     <rest:header name="Cache-Control"
8       value="no-cache,no-store,max-age=0,must-revalidate"/>
9     <rest:header name="X-Content-Type-Options" value="nosniff"/>
10    <rest:header name="Pragma" value="no-cache"/>
11    <rest:header name="Expires" value="0"/>
12    <rest:header name="X-XSS-Protection" value="1;mode=block"/>
13    <rest:header name="Content-Length" value="0"/>
14    <rest:header name="X-AE-REDIRECTION-COUNT" value="2"/>
15    <rest:header name="Date" value="Tue, 21 Feb 2017 10:38:41 GMT"/>
16    <rest:header name="X-AE-HTTP-EXECUTION-TIME-IN-MILLIS" value="42"/>
17  </rest:headers>
18 </rest:RESTResponse>
```

## オプション 2: ファイルインポートによるサービスコネクタの生成

WSDL (Web Service Definition Language) ファイル、Swagger 2.0 JSON ファイル、OpenAPI 3.0 JSON ファイルまたは YAML ファイルをインポートして、アプリケーションの統合サービスコネクタを作成できます。

### WSDL ファイルからのサービスコネクタの作成

WSDL ファイルをインポートして、複数の操作を行うサービスコネクタを容易に作成することができます。

WSDL ファイルをインポートしてサービスコネクタを作成するには、**[WSDL からの新規サービスコネクタ]** ウ

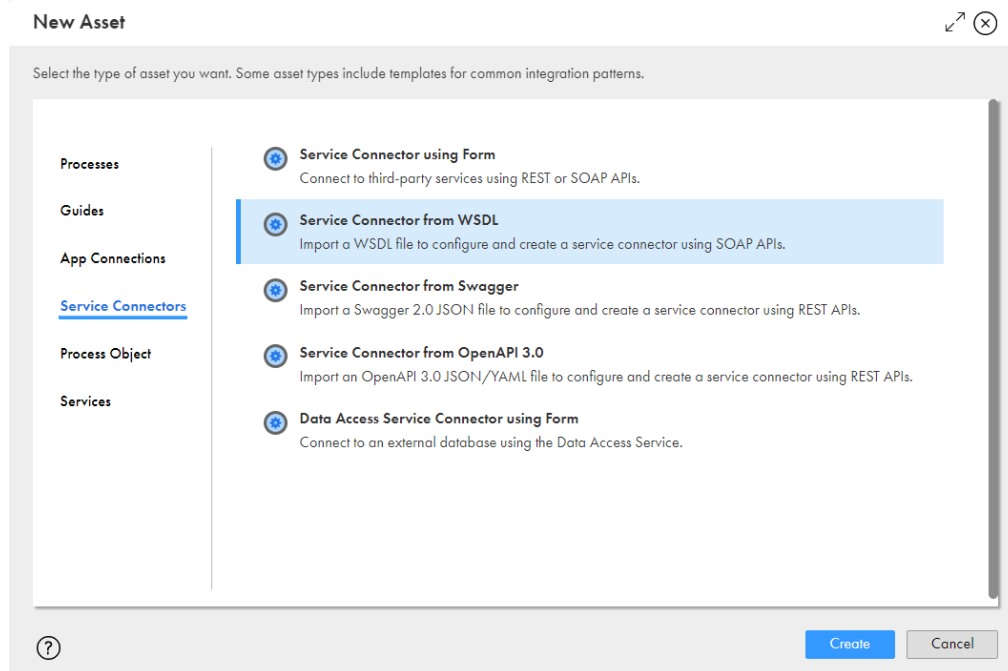
ィザードを使用します。WSDL ファイルを指定すると、アプリケーション統合によってファイルが解析され、操作がロードされます。

インポート中にエラーが発生すると、ウィザードの **【進行状況】** タブの **【エラー/警告】** にエラーが表示されます。エラーに対処したら、WSDL ファイルを再びインポートできます。

1. アプリケーションの統合の左側のナビゲーションバーにある **【新規】** をクリックします。

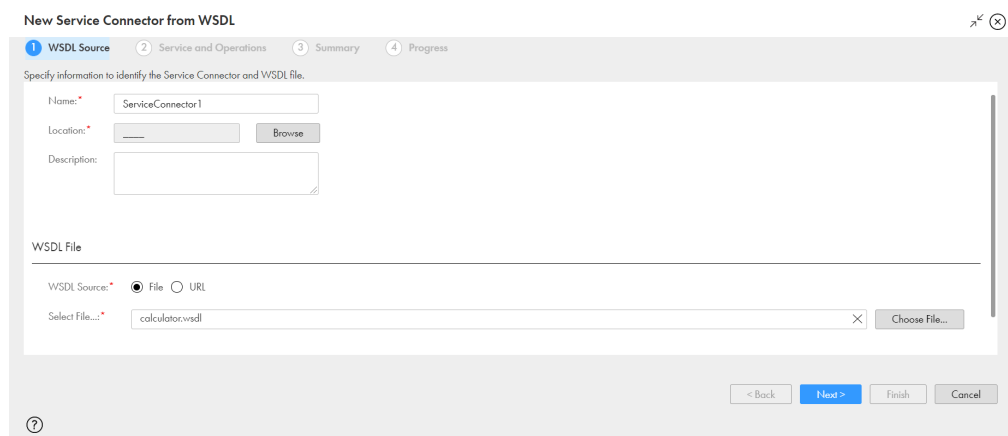
**【新しいアセット】** ダイアログボックスが表示されます。

次の図は、**【新しいアセット】** ダイアログボックスを示しています。



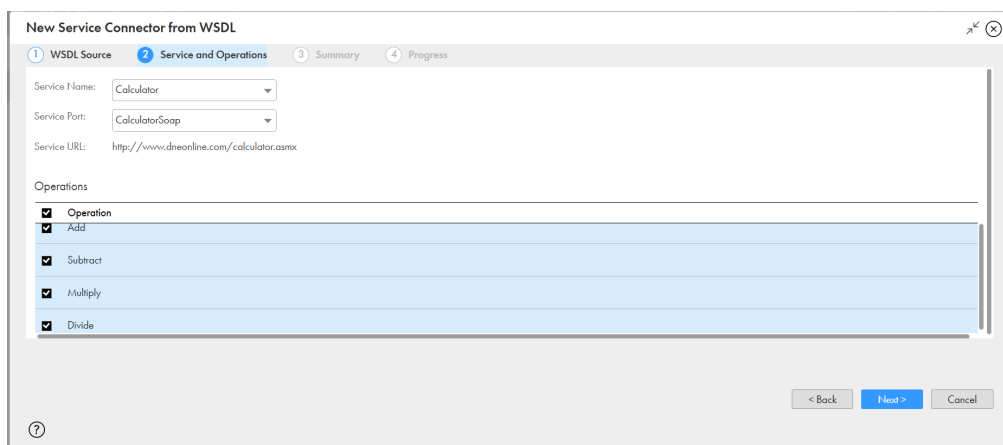
2. **【サービスコネクタ】** > **【WSDL のサービスコネクタ】** をクリックし、**【作成】** をクリックします。

次の図に示すように、**【WSDL ソース】** タブがある **【WSDL からの新規サービスコネクタ】** ダイアログボックスが表示されます。



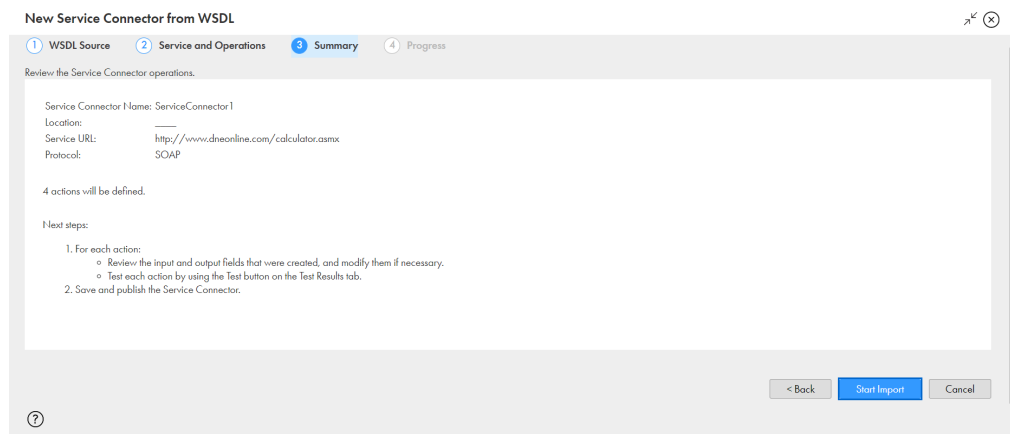
3. サービスコネクタの名前、場所、説明を指定します。
4. **【WSDL ファイル】** セクションで、次のいずれかのオプションを使用して、インポートする WSDL ファイルを指定します。

- ローカルシステムの WSDL ファイルを使用するには、**【ファイル】** を選択してから WSDL ファイルを選択します。  
自己完結型の .wsdl ファイルと .xsd スキーマファイルを含む .wsdl ファイルまたは .zip ファイルを 1 つ 選択できます。
  - WSDL ファイルを含む URL を使用するには、**【URL】** を選択します。**【WSDL URL】** フィールドと **【認証の使用】** フィールドが表示されます。WSDL ファイルを含む URL を **【WSDL URL】** フィールドに入力します。**【認証の使用】** オプションを選択して、WSDL URL にアクセスするためのユーザー名とパスワードを指定します。
5. **【次へ】** をクリックします。  
**【サービスと操作】** タブが表示されます。  
次の図は、**【サービスと操作】** タブを示します。



6. インポートした WSDL ファイルから使用するサービスを **【サービス名】** リストから選択します。  
サービスコネクタが要求を送信する URL が **【サービス URL】** フィールドに表示されます。この URL は編集できません。
7. サービスコネクタが使用するポートを **【サービスポート】** リストから選択します。
8. **【操作】** セクションで、サービスコネクタで提供する操作を選択します。  
選択する操作は、プロセスを作成し、サービスコネクタを使用するときに、アクションとして使用できます。
9. **【次へ】** をクリックします。  
**【サマリ】** タブが表示されます。

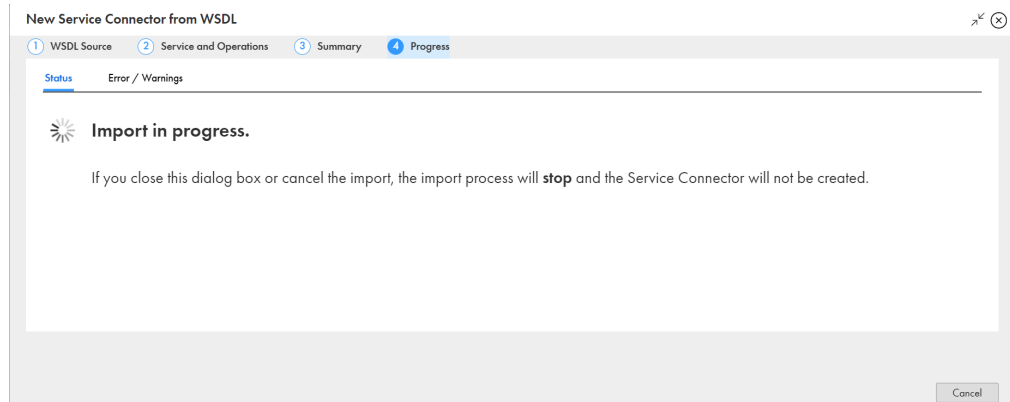
次の図は、【サマリ】タブを示しています。



10. サービスコネクタの設定を確認します。
11. 【インポートの開始】をクリックします。

アプリケーション統合によって WSDL ファイルのインポートが開始され、ステータスが【進行状況】タブに表示されます。【進行状況】タブは、インポートが正常に完了したか、失敗したか、警告が伴う状態で完了したかを示します。

次の図は、【進行状況】タブを示しています。

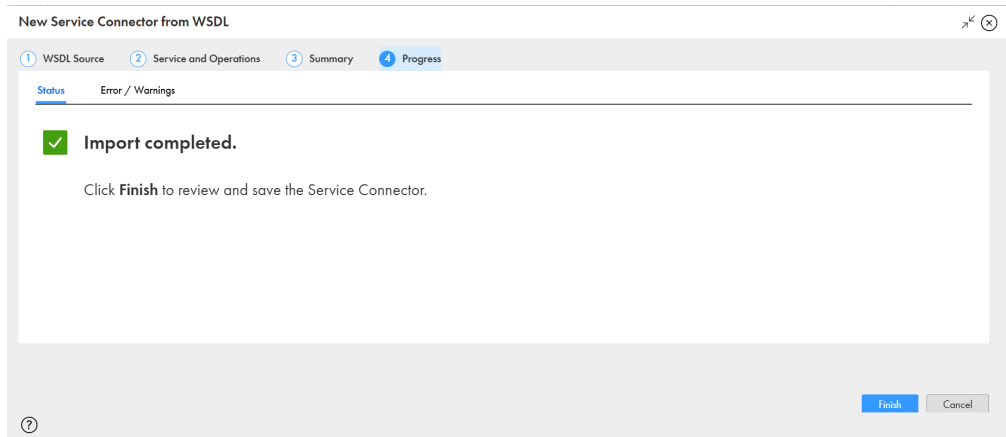


**注:** ダイアログボックスを閉じるかインポートをキャンセルすると、インポートプロセスが停止し、サービスコネクタは作成されません。

インポートが失敗するか、警告を伴う状態で完了した場合は、【エラー/警告】タブをクリックして、インポートが正常に完了しなかった理由を確認します。問題を修正したら、WSDL ファイルを再びインポートできます。

次の図は、成功したインポートを示しています。





12. 【完了】をクリックしてサービスコネクタを保存します。

## 修飾要素が含まれる WSDL ファイル

ユーザー定義のデータ型を持つ WSDL ファイルがある場合、名前空間のプレフィックスと型名を追加して、修飾要素が含まれる WSDL ファイルを作成できます。この WSDL ファイルを使用して、サービスコネクタを作成できます。

ユーザー定義のデータ型が含まれる WSDL ファイルからサービスコネクタを作成できます。名前空間のプレフィックスと型名をユーザー定義のデータに追加して、修飾要素が含まれる WSDL ファイルを作成します。この WSDL ファイルを使用して、サービスコネクタを生成できます。名前空間のプレフィックスは、プロセスオブジェクトには表示されずに、正しいペイロードが表示されます。

例として、ユーザー定義の要素 RecordId と AddressComplete が含まれる WSDL ファイルを考えます。

まず、型名「Address」を持つスキーマを定義します。次に示すのは、complexType 名 Address を持つサンプルスキーマです。

```

xs:complexType name="Address">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="RecordId" type="xs:string" />
    <xs:element minOccurs="0" maxOccurs="1" name="AddressComplete" type="xs:string" />
  </xs:sequence>
</xs:complexType>
<s:element minOccurs="0" maxOccurs="1" name="Address" type="tns:Address" />

```

次に、WSDL ファイルで、名前空間プレフィックス「tns」を次のサンプルのように追加します。

```

<wsdl:message name="InputRequest">
  <wsdl:part name="parameters" element="tns:Address" />
</wsdl:message>
<wsdl:portType name="AddressValidationSoap">
  <wsdl:operation name="Process">
    <wsdl:input message="tns:InputRequest" />
    ...
  </wsdl:operation>
</wsdl:portType>

```

この WSDL ファイルを使用して、サービスコネクタを作成できます。2つのフィールド RecordId と AddressComplete を持つ Address プロセスオブジェクトを取得します。

## 繰り返し要素が含まれる WSDL ファイル

ネストした繰り返し要素を持つ WSDL ファイルからサービスコネクタを作成できます。

例えば、ネストした繰り返し要素の「shelf」と「book」を持つ、次の WSDL と WSDL スキーマファイルを考えます。

```
1 <?xml version="1.0" encoding="UTF-8" standalone="no" ?>
2 <wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
3   xmlns:libSchema="www.library.com/library.xsd"
4   targetNamespace="www.library.com/library.wsdl"
5   xmlns:lib="www.library.com/library.wsdl"
6   xmlns:wSDL="http://schemas.xmlsoap.org/wsdl/"
7   xmlns:xsd="http://www.w3.org/2001/XMLSchema"
8   name="library">
9
10  <wsdl:types>
11    <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
12      <xsd:import
13        namespace="www.library.com/library.xsd"
14        schemaLocation="./schema/library.xsd">
15      </xsd:import>
16    </xsd:schema>
17  </wsdl:types>
18
19  <wsdl:message name="request">
20    <wsdl:part name="request" element="libSchema:library" />
21  </wsdl:message>
22  <wsdl:message name="response">
23    <wsdl:part name="response" element="libSchema:bookCount" />
24  </wsdl:message>
25
26  <wsdl:portType name="Library">
27    <wsdl:operation name="countBooks">
28      <wsdl:input message="lib:request" />
29      <wsdl:output message="lib:response"></wsdl:output>
30    </wsdl:operation>
31  </wsdl:portType>
32
33 </wsdl:definitions>
34
```

```
1 <?xml version="1.0"?>
2 <xs:schema elementFormDefault="qualified"
3   targetNamespace="www.library.com/library.xsd"
4   xmlns:xs="http://www.w3.org/2001/XMLSchema"
5   xmlns:ns="www.library.com/library.xsd">
6
7   <xs:element name="book">
8     <xs:complexType>
9       <xs:attribute name="name" type="xs:string" use="required" />
10    </xs:complexType>
11  </xs:element>
12  <xs:element name="shelf">
13    <xs:complexType>
14      <xs:sequence>
15        <xs:element ref="ns:book" minOccurs="0" maxOccurs="100" />
16      </xs:sequence>
17      <xs:attribute name="name" type="xs:string" use="required" />
18    </xs:complexType>
19  </xs:element>
20  <xs:element name="library">
21    <xs:complexType>
22      <xs:sequence>
23        <xs:element ref="ns:shelf" minOccurs="0" maxOccurs="20" />
24      </xs:sequence>
25      <xs:attribute name="name" type="xs:string" use="required" />
26    </xs:complexType>
27  </xs:element>
28
29  <xs:element name="bookCount" type="xs:integer" />
30
31 </xs:schema>
```

この WSDL と WSDL スキーマファイルを使用して、WSDL からサービスコネクタツールによってサービスコネクタを作成できます。サービスコネクタをインポートすると、Library、Shelf、および Book の 3 つのプロセスオブジェクトがあります。親オブジェクトを選択すると、下部パネルの右側にネストされたオブジェクトも表示されます。

## elementFormDefault

WSDL スキーマの elementFormDefault 属性が含まれる WSDL ファイルを使用できます。

WSDL ファイルには、elementFormDefault="qualified" または elementFormDefault="unqualified" を含むことができます。

## 選択要素

選択スキーマ要素を使用して、WSDL ファイル内にパイロードを作成できます。

if 要素を使用して、要求に表示すべきものを示します。

サービスコネクタへの WSDL は、最上位レベルの選択要素を考慮します。これは、WSDL スキーマファイルにネストした選択要素がある場合、サービスコネクタへの WSDL が最初の選択要素を取得します。

## Swagger JSON ファイルからのサービスコネクタの作成

Swagger 2.0 JSON ファイルをインポートして、複数の操作を行うサービスコネクタを容易に作成することができます。Swagger JSON ファイルをインポートしてサービスコネクタを作成するには、**Swagger からの新**

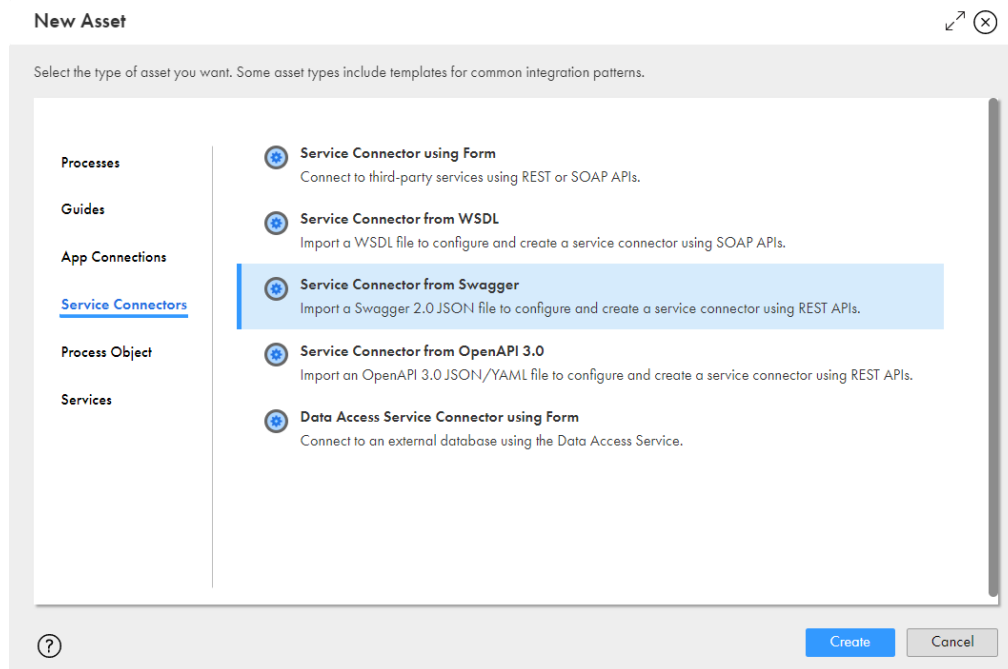
**規サービスコネクタ** ウィザードを使用します。Swagger JSON ファイルを指定すると、アプリケーション統合によってファイルが解析され、操作がロードされます。

インポート中にエラーが発生すると、ウィザードの **【進行状況】** タブの **【エラー/警告】** にエラーが表示されます。エラーに対処したら、Swagger JSON ファイルを再びインポートできます。

1. アプリケーションの統合の左側のナビゲーションバーにある **【新規】** をクリックします。

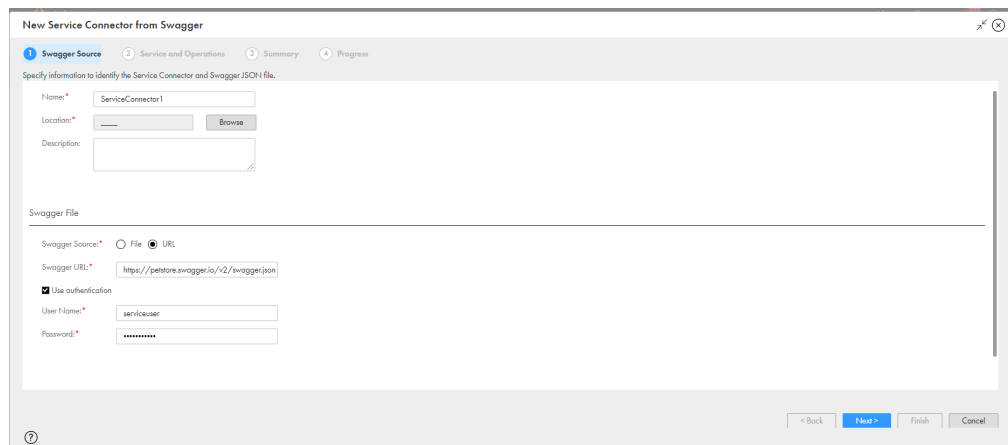
**【新しいアセット】** ダイアログボックスが表示されます。

次の図は、**【新しいアセット】** ダイアログボックスを示しています。



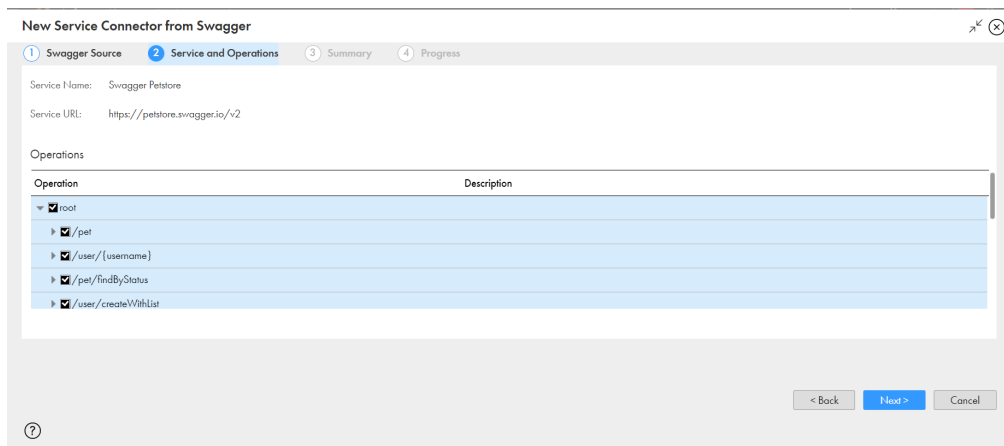
2. **【サービスコネクタ】** > **【Swagger のサービスコネクタ】** をクリックし、**【作成】** をクリックします。

次の図に示すように、**【Swagger ソース】** タブがある **【Swagger からの新規サービスコネクタ】** ダイアログボックスが表示されます。

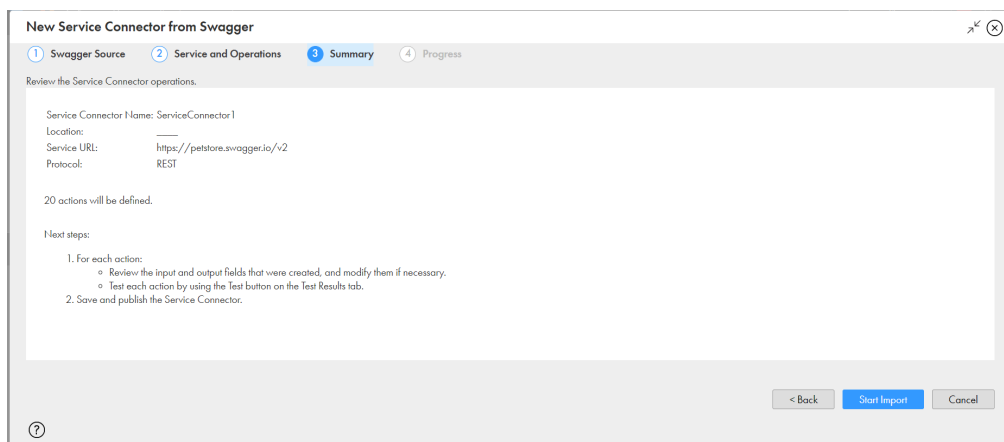


3. サービスコネクタの名前、場所、説明を指定します。
4. **【Swagger ファイル】** セクションで、次のいずれかのオプションを使用して、インポートする Swagger JSON ファイルを指定します。

- ローカルシステムの Swagger JSON ファイルを使用するには、**[ファイル]** を選択してから Swagger JSON ファイルを選択します。
  - Swagger JSON ファイルが含まれる URL を使用するには、**[URL]** を選択します。**[Swagger URL]** フィールドと **[認証の使用]** フィールドが表示されます。Swagger JSON ファイルを含む URL を **[Swagger URL]** フィールドに入力します。**[認証の使用]** オプションを選択して、Swagger JSON URL にアクセスするためのユーザー名とパスワードを指定します。
5. **[次へ]** をクリックします。
- [サービスと操作]** タブが表示されます。
- 次の図は、**[サービスと操作]** タブを示します。



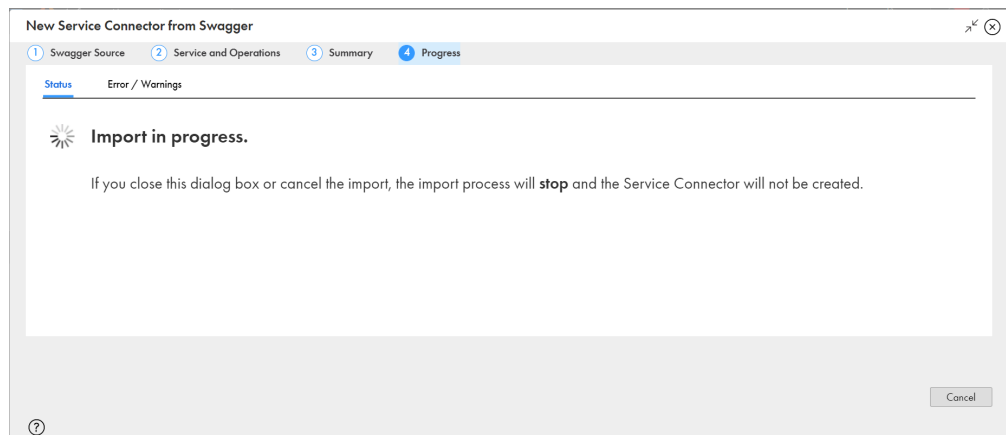
6. **[操作]** セクションで、サービスコネクタで提供する操作を選択します。
- 選択する操作は、プロセスを作成し、サービスコネクタを使用するときに、アクションとして使用できます。
7. **[次へ]** をクリックします。
- [サマリ]** タブが表示されます。
- 次の図は、**[サマリ]** タブを示しています。



8. サービスコネクタの設定を確認します。
9. **[インポートの開始]** をクリックします。

アプリケーション統合によって Swagger JSON ファイルのインポートが開始され、ステータスが **【進行状況】** タブに表示されます。**【進行状況】** タブは、インポートが正常に完了したか、失敗したか、警告が伴う状態で完了したかを示します。

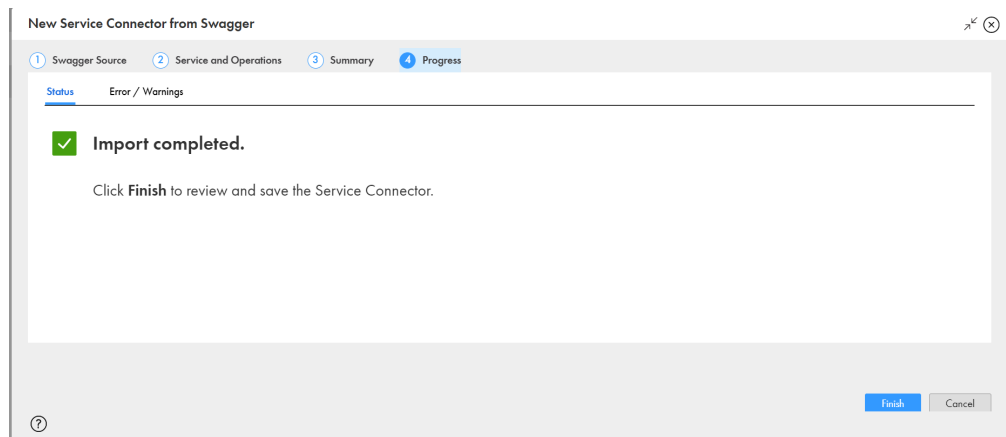
次の図は、**【進行状況】** タブを示しています。



**注:** ダイアログボックスを閉じるかインポートをキャンセルすると、インポートプロセスが停止し、サービスコネクタは作成されません。

インポートが失敗するか、警告を伴う状態で完了した場合は、**【エラー/警告】** タブをクリックして、インポートが正常に完了しなかった理由を確認します。問題を修正したら、Swagger JSON ファイルを再びインポートできます。

次の図は、成功したインポートを示しています。



10. **【完了】** をクリックしてサービスコネクタを保存します。

## Swagger ファイルに関するルールおよびガイドライン

Swagger ファイルをインポートしてサービスコネクタを作成する場合は、次のルールとガイドラインを考慮してください。

- 有効な Swagger ファイルをインポートする必要があります。次の URL を使用して、Swagger ファイルの構文を検証できます。  
<https://editor.swagger.io/>
- アプリケーション統合は、JSON 形式の Swagger ファイルをサポートしています。YAML などの他の形式はサポートしていません。

- アプリケーション統合は、単一のドキュメントの Swagger ファイルのみをサポートします。マルチドキュメントの Swagger ファイルをインポートする場合、アプリケーション統合は依存関係をロードしません。
- 複雑なタイプおよび操作が多数含まれた Swagger ファイルをインポートしてサービスコネクタを作成すると、生成されるサービスコネクタが複雑化してサイズも大きくなるため、サービスコネクタエディタに影響を及ぼす可能性があります。
- 複数のタイプと操作を含んだサービスコネクタを作成するための大きな Swagger ファイルがある場合は、Swagger ファイルを複数の小さな Swagger ファイルに分割して、パフォーマンスと使いやすさを向上させることができます。Swagger 分割ツールを使用して、Swagger ファイルを分割できます。詳細については、次のコミュニティの記事を参照してください：  
<https://knowledge.informatica.com/s/article/DOC-18606>  
インポートウィザードで操作のサブセットを手動で選択することで、より少ないアクションでサービスコネクタを作成することもできます。

## OpenAPI 3.0 ファイルからのサービスコネクタの作成

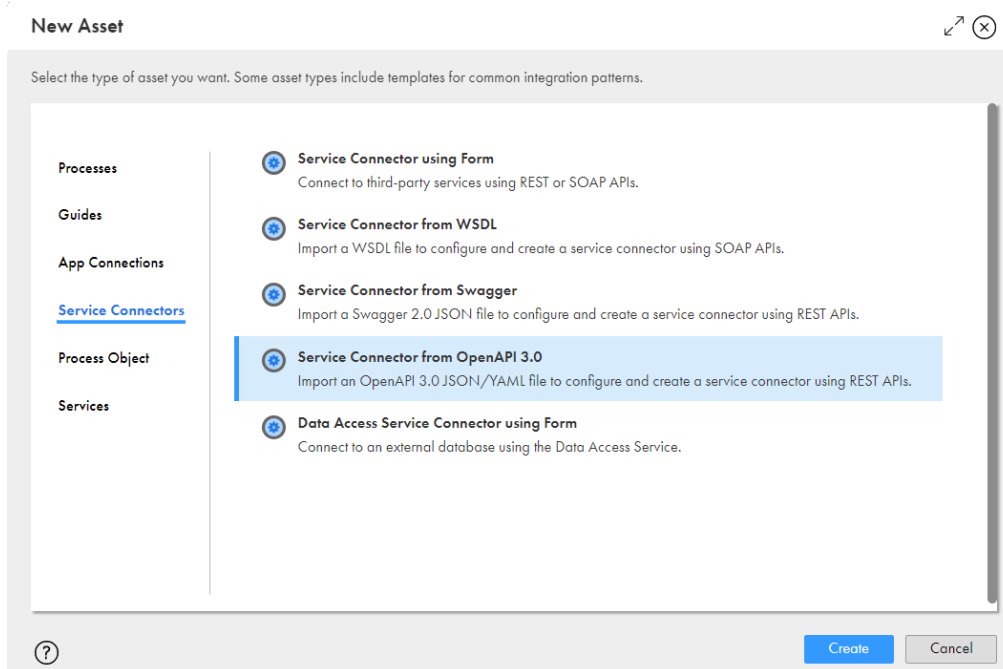
OpenAPI 3.0 JSON ファイルまたは YAML ファイルをインポートして、複数の操作を行うサービスコネクタを容易に作成することができます。OpenAPI 3.0 ファイルをインポートしてサービスコネクタを作成するには、**[OpenAPI 3.0 からの新規サービスコネクタ]** ウィザードを使用します。OpenAPI 3.0 ファイルを指定すると、アプリケーション統合によってファイルが解析され、操作がロードされます。

インポート中にエラーが発生すると、ウィザードの **[進行状況]** タブの **[エラー/警告]** にエラーが表示されます。エラーに対処したら、OpenAPI 3.0 ファイルを再びインポートできます。

1. アプリケーションの統合の左側のナビゲーションバーにある **[新規]** をクリックします。

**[新しいアセット]** ダイアログボックスが表示されます。

次の図は、**[新しいアセット]** ダイアログボックスを示しています。



2. **[サービスコネクタ]** > **[OpenAPI 3.0 からのサービスコネクタ]** をクリックし、**[作成]** をクリックします。

次の図に示すように、[OpenAPI 3.0 ソース] タブが表示された [OpenAPI 3.0 からの新規サービスコネクタ] ダイアログボックスが表示されます。

New Service Connector from OpenAPI 3.0

1 OpenAPI 3.0 Source 2 Service and Operations 3 Summary 4 Progress

Specify information to identify the Service Connector and OpenAPI 3.0 JSON/YAML file.

Name:\* ServiceConnector1

Location:\* Default Browse

Description:

OpenAPI 3.0 File

OpenAPI 3.0 Source:\*  File  URL

Select File...: Drop file here Choose File...

< Back Next > Finish Cancel

?

3. サービスコネクタの名前、場所、説明を指定します。
4. **[OpenAPI 3.0 ファイル]** セクションで、次のいずれかのオプションを使用して、インポートする OpenAPI 3.0 ファイルを指定します。
  - ローカルシステムの OpenAPI 3.0 ファイルを使用するには、**[ファイル]** を選択してから OpenAPI 3.0 JSON ファイルまたは YAML ファイルを選択します。  
単一の .json ファイルか .yaml ファイル、または、自己完結型の .json ファイルか .yaml ファイルを含む単一の .zip ファイルを選択することもできます。
  - OpenAPI 3.0 ファイルを含む URL を使用するには、**[URL]** を選択します。**[OpenAPI 3.0 URL]** フィールドと **[認証の使用]** フィールドが表示されます。**[OpenAPI 3.0 URL]** フィールドに、OpenAPI 3.0 JSON ファイルまたは YAML ファイルを含む URL を入力します。**[認証の使用]** オプションを選択して、OpenAPI 3.0 URL にアクセスするためのユーザー名とパスワードを指定します。
5. **[次へ]** をクリックします。  
**[サービスと操作]** タブが表示されます。  
次の図は、**[サービスと操作]** タブを示しています。



New Service Connector from OpenAPI 3.0 ↗ ⓧ

① OpenAPI 3.0 Source   ② Service and Operations   ③ Summary   ④ Progress

Service Name: Swagger Petstore - OpenAPI 3.0

Service URL:

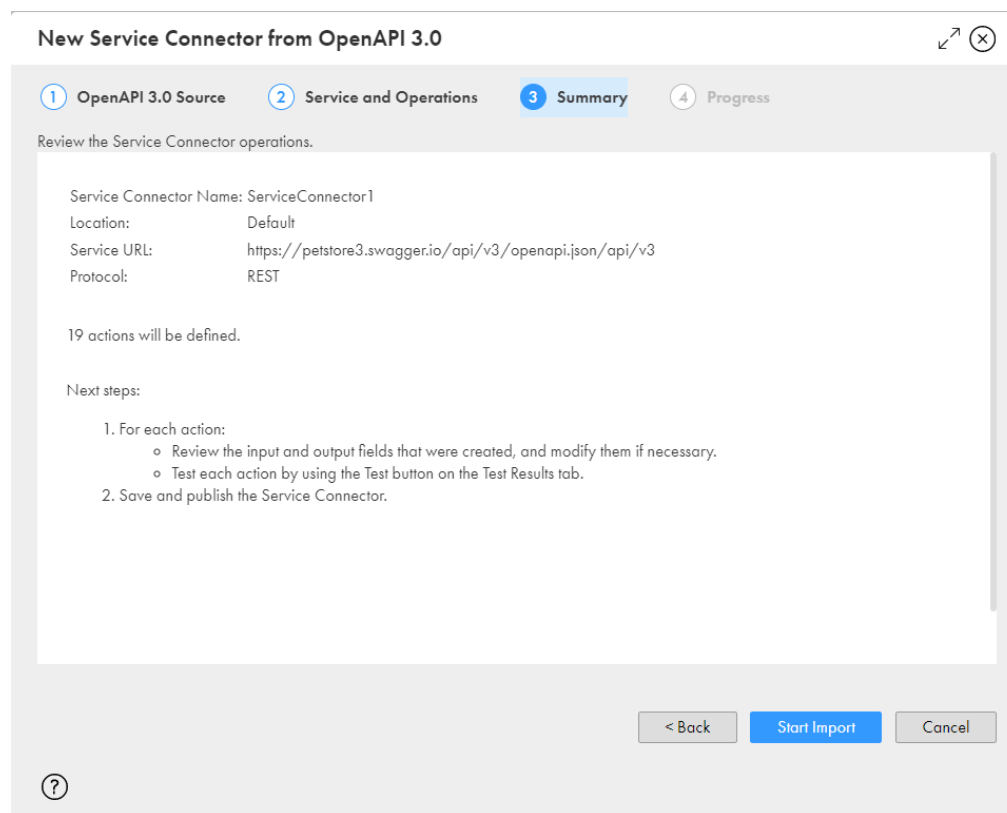
Operations

Operation	Description
<input checked="" type="checkbox"/> root	

6. **【サービス URL】** リストで、インポートした OpenAPI 3.0 ファイルから使用するエンドポイントを選択して、サービスコネクタを作成します。
7. **【操作】** セクションで、サービスコネクタで提供する操作を選択します。  
選択する操作は、プロセスを作成し、サービスコネクタを使用するときに、アクションとして使用できません。
8. **【次へ】** をクリックします。  
**【サマリ】** タブが表示されます。

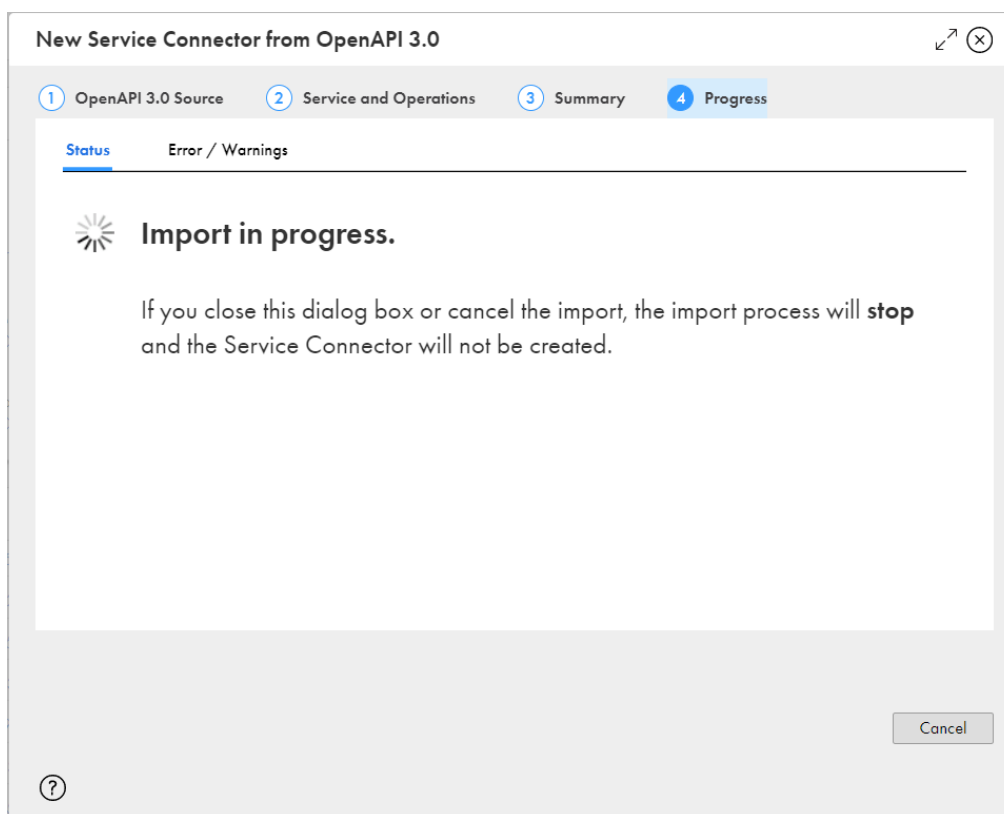
次の図は、[サマリ] タブを示しています。



9. サービスコネクタの設定を確認します。
10. **[インポートの開始]** をクリックします。

アプリケーション統合によって OpenAPI 3.0 ファイルのインポートが開始され、ステータスが **[進行状況]** タブに表示されます。**[進行状況]** タブは、インポートが正常に完了したか、失敗したか、警告が伴う状態で完了したかを示します。

次の図は、**[進行状況]** タブを示しています。



**注:** ダイアログボックスを閉じるかインポートをキャンセルすると、インポートプロセスが停止し、サービスコネクタは作成されません。

インポートが失敗するか、警告を伴う状態で完了した場合は、[エラー/警告] タブをクリックして、インポートが正常に完了しなかった理由を確認します。問題を修正したら、OpenAPI 3.0 ファイルを再びインポートできます。

11. **【完了】** をクリックしてサービスコネクタを保存します。

## HTTP 操作

DELETE、PUT、POST、PATCH、および GET HTTP の各操作が含まれる Swagger ファイルをインポートできます。

生成するサービスコネクタは、Swagger ファイルの HTTP 操作に対応するアクションを持ちます。Swagger ファイルに複数の HTTP 操作がある場合、サービスコネクタは複数のアクションを持ちます。

例えば、POST HTTP 操作が 1 つある Swagger ファイルを使用する場合、得られるのは POST アクションが 1 つあるサービスコネクタです。

次の図は、Swagger ファイル内の 1 つの POST HTTP 操作を示します。

```

- paths: {
  - /pet: {
    - post: {
      - tags: [
        "pet"
      ],
      summary: "Add a new pet to the store",
      description: "",
      operationId: "addPet",
      - consumes: [
        "application/json",
        "application/xml"
      ],
      - produces: [
        "application/xml",
        "application/json"
      ],
      - parameters: [
        - {
          in: "body",
          name: "body",
          description: "Pet object that needs to be added to the store",
          required: true,
          - schema: {
            $ref: "#/definitions/Pet"
          }
        }
      ],
      - responses: {
        - 405: {
          description: "Invalid input"
        }
      },
      - security: [
        - {
          - petstore_auth: [
            "write:pets",
            "read:pets"
          ]
        }
      ]
    }
  }
}

```

次の図は、先行する Swagger ファイルを使用するとき取得する、サービスコネクタのアクションを示します。

The screenshot shows the Swagger UI interface. The top part is a table of actions:

Action Name	Action Type	Description
addPet	general	Add a new pet to the store
updatePet	general	Update an existing pet
getUserByName	general	Get user by user name
deleteUser	general	Delete user. This can only be done by the logged in user.
updateUser	general	Updated user. This can only be done by the logged in user.

The bottom part shows the configuration for the 'addPet' action in the 'Binding' tab:

- URL\*:
- Verb:
- Multi Using:
- Authentication Type:
- Other Parameters:
- HTTP Headers:

Add Pet アクションが動詞 POST に対応しています。

## JSON ペイロード

異なるタイプの JSON ペイロードを持つ Swagger ファイルをインポートできます。生成するサービスコネクタは、Swagger ファイルの JSON ペイロードを反映します。

次の表は、使用できる JSON ペイロード機能、Swagger サンプルの抜粋、および対応するサービスコネクタの入力フィールドを示します。

JSON ペイロードの機能	サンプル Swagger コンテンツ	サービスコネクタの入力フィールド
オブジェクトのリスト	<pre>schema: {   type: "array",   - items: {     \$ref: "#/definitions/User"   } }</pre>	
参照	<pre>schema: {   \$ref: "#/definitions/Pet" }</pre>	
単一の選択列挙	<pre>items: {   type: "string",   - enum: [     "available",     "pending",     "sold"   ],   default: "available" },</pre>	
単純型	<pre>{   name: "orderId",   in: "path",   description: "ID of pet that needs to be fetched",   required: true,   type: "integer",   maximum: 10,   minimum: 1,   format: "int32" }</pre>	
添付ファイル	<pre>- {   name: "file",   in: "formData",   description: "file to upload",   required: false,   type: "file" }</pre>	
単純型のリスト	<pre>- parameters: [   {     name: "tags",     in: "query",     description: "Tags to filter by",     required: true,     type: "array",     - items: {       type: "string"     },     collectionFormat: "multi"   } ],</pre>	
複数の選択列挙	<pre>parameters: [   {     name: "status",     in: "query",     description: "status values that need to be considered for filter",     required: true,     type: "array",     - items: [       {         type: "string",         - enum: [           "available",           "pending",           "sold"         ],         default: "available"       }     ],     collectionFormat: "multi"   } ],</pre>	
他のオブジェクトリストへの参照付きオブジェクト	<pre>tags: {   type: "array",   - xml: {     name: "tag",     wrapped: true   },   - items: {     \$ref: "#/definitions/Tag"   } },</pre>	

## 必須要素と NULL 可能要素

ファイルをインポートしてサービスコネクタを生成する前に、WSDL ファイル、Swagger JSON ファイル、および OpenAPI 3.0 JSON ファイルまたは YAML ファイルで、プロセスオブジェクトフィールドを必須および NULL 可能として定義できます。

WSDL ファイルで、フィールドを NULL 可能として定義するには、フィールド要素で `nillable` 属性を `"true"` に設定します。`nillable` 属性を指定しない場合、デフォルトでは、フィールドは NULL 可能ではないものとして設定されます。フィールドを必須として設定しない場合は、フィールド要素で `minOccurs` 属性を `"0"` に設定します。`minOccurs` 属性を指定しない場合、デフォルトでは、フィールドは必須と見なされます。

例えば、`ProcessObjectReq` という名前のスキーマを持つ WSDL ファイルがあるとします。このスキーマは複合型で、`Name` と `ID` という 2 つの要素を含んでいます。ID フィールドを NULL 可能とし、必須としない場合は、次のサンプルに示すように、`nillable` 属性を `"true"` に設定し、`minOccurs` 属性を `"0"` に設定します。

```
<xsd:complexType name="ProcessObjectReq">
  <xsd:sequence>
    <xsd:element name="Name" nillable="false" type="xsd:string"/>
    <xsd:element name="ID" minOccurs="0" nillable="true" type="xsd:integer"/>
  </xsd:sequence>
</xsd:complexType>
```

WSDL ファイルをインポートすると、次の図に示すように、WSDL ファイルで定義された NULL 可能および必須のフィールド要素がプロセスオブジェクトにプロパゲートされます。

Field Name*	Type	Required	Nullable
Name	Text	<input checked="" type="checkbox"/>	<input type="checkbox"/>
ID	Number	<input type="checkbox"/>	<input checked="" type="checkbox"/>

同様に、次のサンプルに示すように、Swagger JSON ファイルでも、必須および NULL 可能なフィールド要素を設定し、サービスコネクタを作成できます。

```
"definitions" : {
  "Process1-3Request" : {
    "type" : "object",
    "required" : [ "Test" ],
    "properties" : {
      "Test" : {
        "$ref" : "#/definitions/ProcessObjectReq"
      }
    }
  },
  "ProcessObjectReq" : {
    "type" : "object",
    "properties" : {
      "Name" : {
        "type" : "string"
      },
      "ID" : {
        "type" : "integer",

```

```

        "format" : "int32"
    }
},
"required": [
    "Name"
],
"x-nullable":["ID"]
}
}
}

```

次のサンプルに示すように、OpenAPI 3.0 ファイルでも、必須および NULL 可能なフィールド要素を設定し、プロセスオブジェクトを作成できます。

```

"ApiResponse": {
  "type": "object",
  "properties": {
    "code": {
      "type": "integer",
      "format": "int32",
      "required": true,
      "nullable": false
    },
  },
  ...
}

```

NULL 可能フィールドがあるプロセスオブジェクトをプロセスで使用している場合に、そのプロセスの Swagger JSON ファイルを生成しても、生成された Swagger JSON ファイルにはプロセスオブジェクトで定義された x-nullable フィールドが含まれないことがあります。デフォルトでは、このフィールドは NULL 可能ではないものとして設定されます。そのため、NULL 可能フィールド値を保持するには、WSDL ファイルまたは OpenAPI 3.0 ファイルを使用することをお勧めします。

次の図は、インポートされた Swagger JSON ファイルからの必須フィールドと NULL 可能フィールドが選択されている状態を示しています：

The screenshot shows the 'ServiceConnector1' interface. At the top, there is a 'Valid' status indicator and a 'Save' button. Below this, the 'Process Objects' tab is active, displaying a list of two objects: 'Process1\_3Request' and 'ProcessObjectReq'. The 'ProcessObjectReq' object is selected, and its 'Fields' are shown in a table below. The table has columns for 'Field Name\*', 'Type', 'Required', and 'Nullable'. The 'Name' field is of type 'Text', is required (checked), and is not nullable. The 'ID' field is of type 'Number', is not required, and is nullable (checked).

Field Name*	Type	Required	Nullable	
Name	Text	<input checked="" type="checkbox"/>	<input type="checkbox"/>	×
ID	Number	<input type="checkbox"/>	<input checked="" type="checkbox"/>	×

このプロセスオブジェクトをプロセスで使用する場合、必須および NULL 可能としてマークされたフィールドは、生成された WSDL ファイル、Swagger ファイル、および OpenAPI 3.0 ファイルで同じ制限を使用します。

# オプション 3 : GitHub からのサービスコネクタのダウンロード

Informatica は複数の REST および SOAP API ベースのコネクタを GitHub にアップロードしました。Informatica サービスコネクタを無料でダウンロードして、組織にインポートできます。サービスコネクタファイルを上アップロードして、リポジトリの充実化に貢献することもできます。

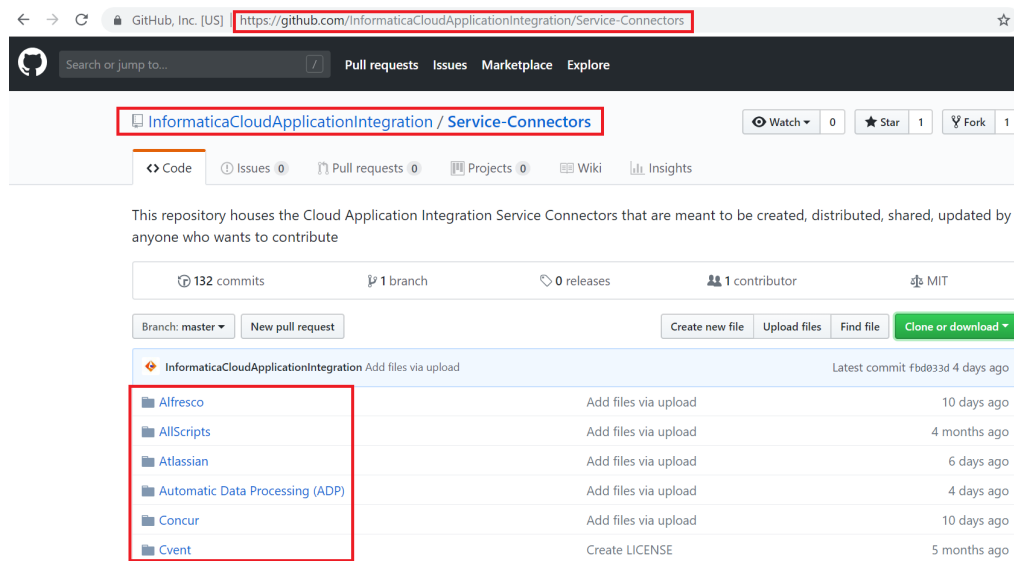
Informatica サービスコネクタとダウンロード可能なプロセスのサンプルについて詳しくは、次の URL をクリックしてください:

<https://network.informatica.com/community/informatica-network/products/cloud-integration/cloud-application-integration/blog/2018/10/16/registry-of-service-connectors-your-gateway-to-building-composite-api-using-cloud-application-integration>

1. 次の URL にアクセスしてサービスコネクタをダウンロードします:

<https://github.com/InformaticaCloudApplicationIntegration/Service-Connectors>

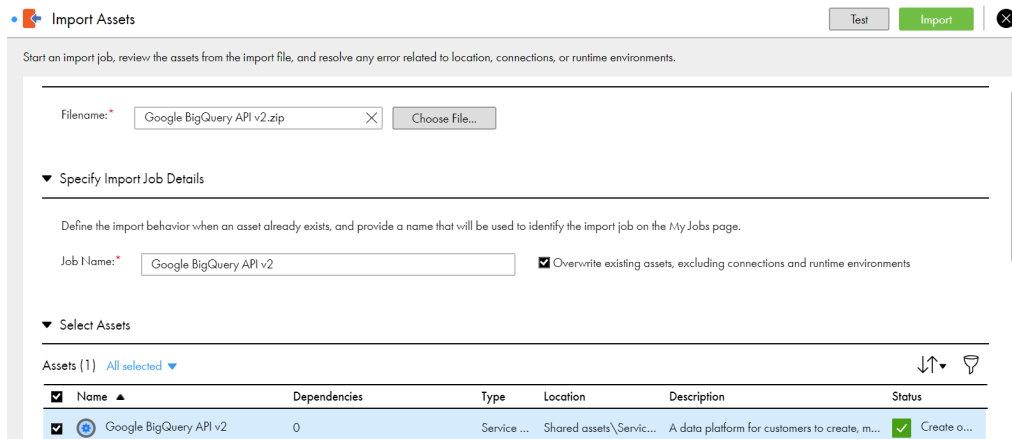
Informatica サービスコネクタが掲載された次の **【サービスコネクタ】** ページが表示されます。



2. 使用するサービスコネクタの .zip ファイルをダウンロードします。
3. Informatica Intelligent Cloud Services<sup>SM</sup>にログインし、**【アプリケーション統合】** をクリックします。
4. **【参照】** ページで **【インポート】** をクリックして、GitHub からダウンロードしたサービスコネクタの .zip ファイルを選択します。

次の図のように、**【アセットの選択】** セクションにサービスコネクタが表示されます。



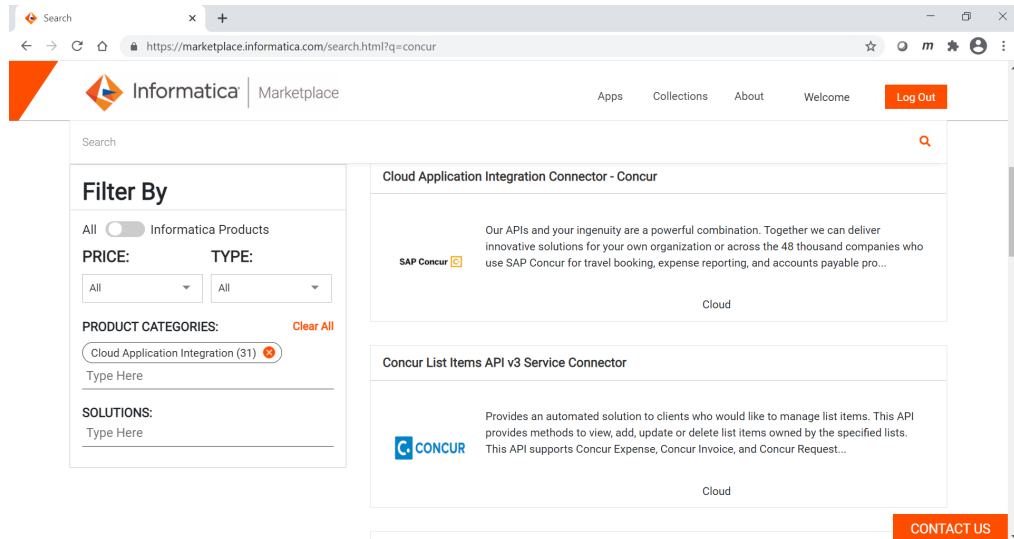


5. **【インポート】** をクリックして **【マイインポート/エクスポートログ】** ページを開き、インポートのステータスを確認します。  
インポートが完了したら、組織内の次の場所でサービスコネクタを利用できます。  
Shared assets\Service Connectors

## オプション 4: Informatica Marketplace からのサービスコネクタのダウンロード

Informatica は、複数の REST および SOAP API ベースのコネクタを Informatica Marketplace にアップロードしています。Informatica Marketplace でアカウントを作成した後に、Informatica サービスコネクタをダウンロードして組織にインポートできます。

1. 次の URL にアクセスします: <https://marketplace.informatica.com/>
2. アカウントを作成してログインします。
3. 左側のパネルの **【製品カテゴリ】** リストから、**【Cloud アプリケーション統合】** をクリックします。  
使用可能なサービスコネクタを示すページが表示されます。  
次の図に、一部のサンプルサービスコネクタを示します。

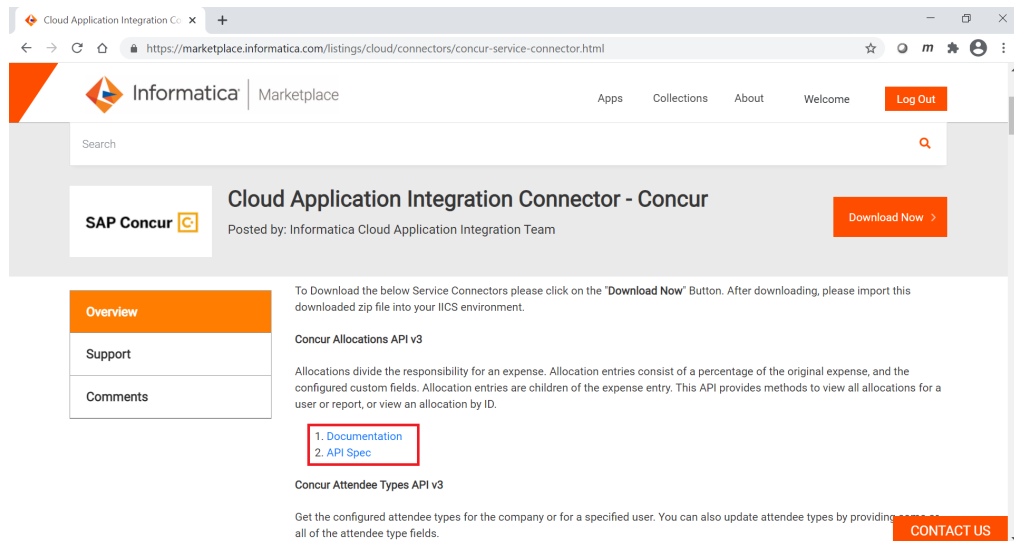


4. **【検索】** ボックスに、ダウンロードするサービスコネクタの名前を入力します。また、**【価格】** リストと**【タイプ】** リストを使用して、サービスコネクタをフィルタリングすることもできます。

5. ダウンロードするサービスコネクタの名前をクリックします。

サービスコネクタの詳細が表示されます。

サービスコネクタに特定の API ベースのサービスコネクタが含まれている場合は、次の図に示すように、すべてのコネクタの詳細と、API ドキュメントおよび API 仕様にアクセスするためのリンクが表示されます。



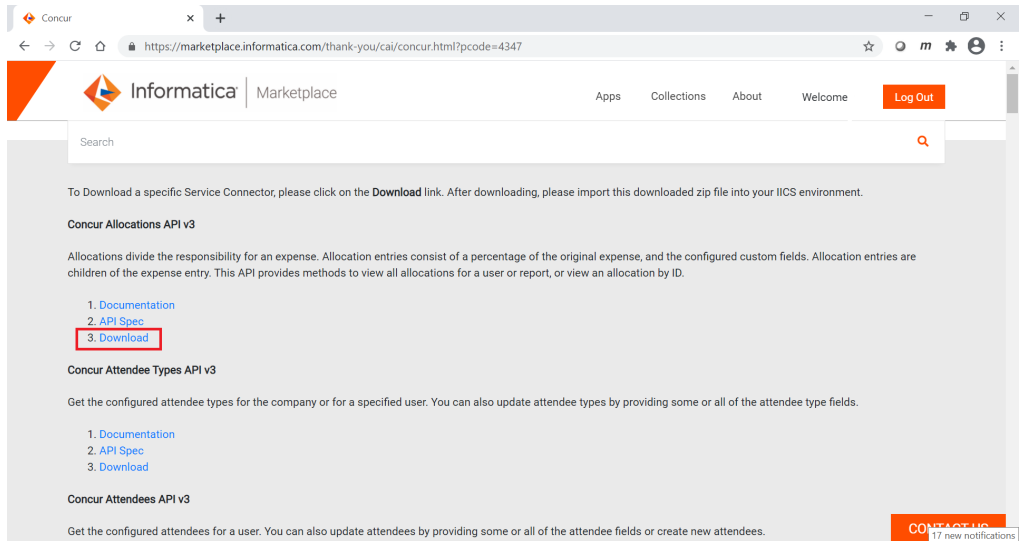
6. **【すぐにダウンロード】** をクリックします。

ユーザーの詳細の入力を求めるフォームが表示されます。

7. フォームに入力し、利用規約に同意して、フォームを送信します。

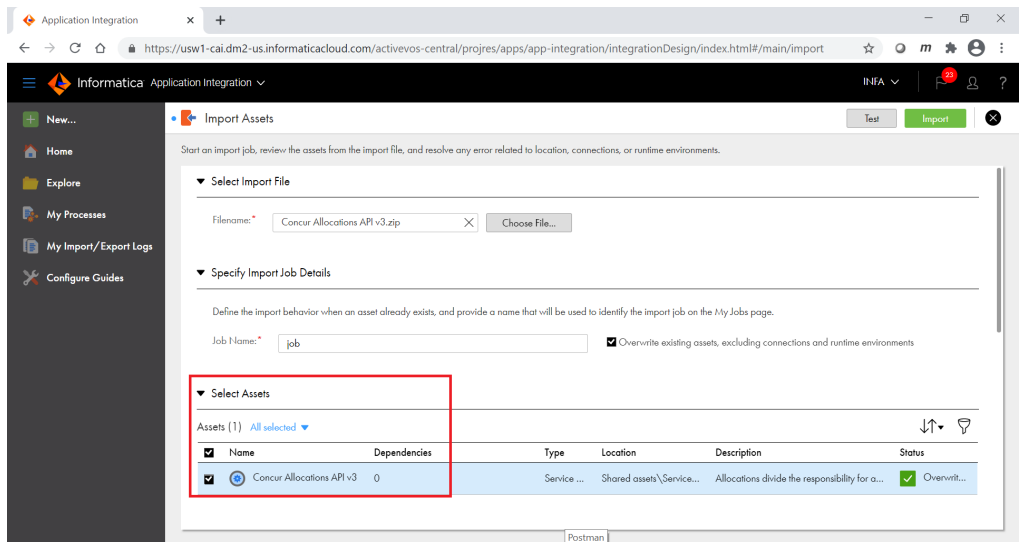
サービスコネクタがダウンロードされます。

サービスコネクタに特定の API ベースのサービスコネクタが含まれている場合は、次の図に示すように、特定のコネクタをダウンロードすることができます。



8. Informatica Intelligent Cloud Services にログインし、**【アプリケーション統合】** をクリックします。
9. **【参照】** ページで **【インポート】** をクリックして、Informatica Marketplace からダウンロードしたサービスコネクタの.zip ファイルを選択します。

次の図のように、**【アセットの選択】** セクションにサービスコネクタが表示されます。



10. **【インポート】** をクリックして **【マイインポート/エクスポートログ】** ページを開き、インポートのステータスを確認します。

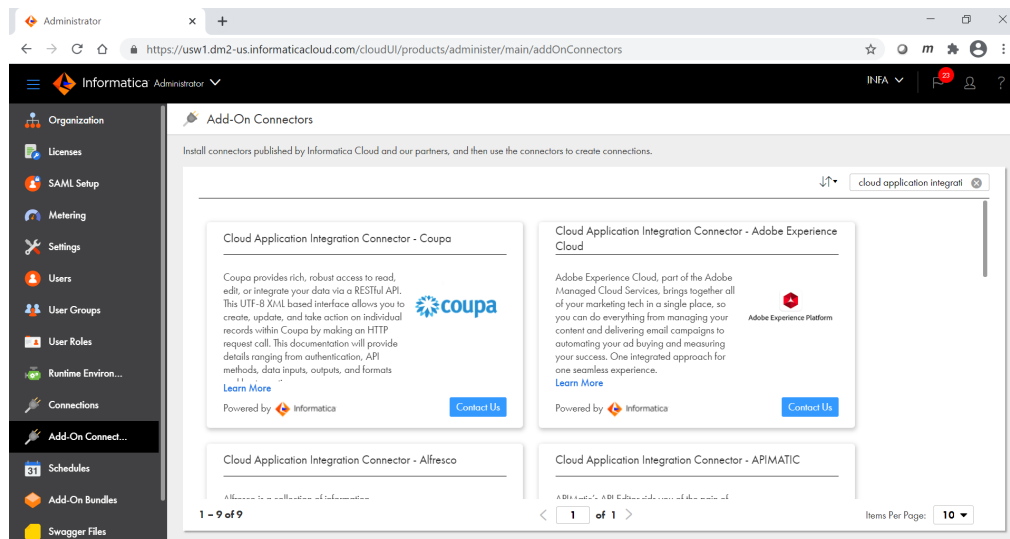
インポートが完了したら、組織内の次の場所でサービスコネクタを利用できます。  
Shared assets\Service Connectors

# オプション 5: Administrator からのサービスコネクタのダウンロード

Informatica Intelligent Cloud Services Administrator から REST および SOAP API ベースのサービスコネクタをダウンロードして、組織にインポートできます。

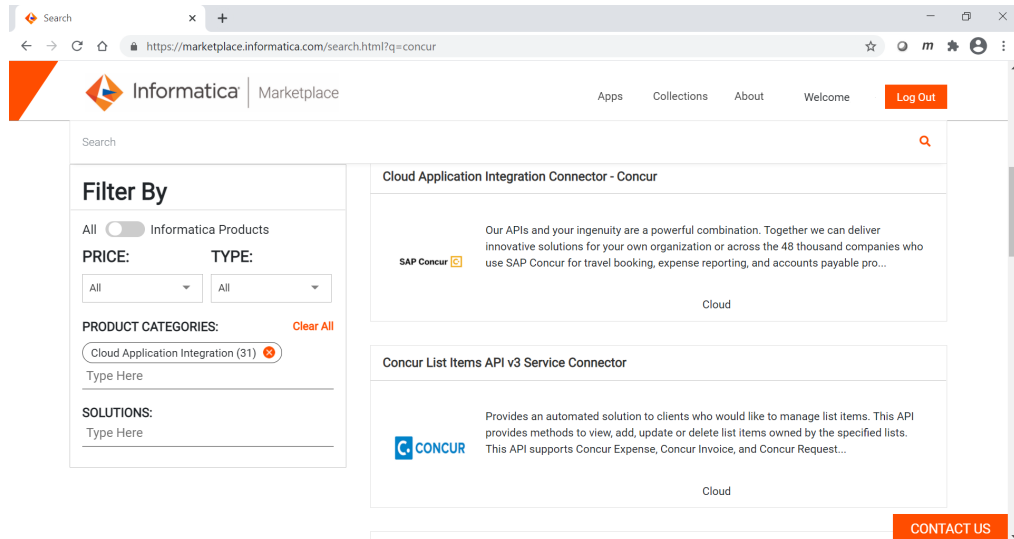
1. Informatica Intelligent Cloud Services にログインし、**[Administrator]** をクリックします。
2. 左側のパネルで、**[アドオンコネクタ]** をクリックします。
3. 右側のパネルで、**[検索]** ボックスに **[Cloud アプリケーション統合コネクタ]** として検索キーワードを入力します。

ダウンロードできる Cloud アプリケーション統合サービスコネクタが、次の図に示すように表示されます。



**注:** Cloud アプリケーション統合サービスコネクタの名前の前には、**[Cloud アプリケーション統合コネクタ]** という語が表示されます。

4. ダウンロードするサービスコネクタに対して **[お問い合わせ]** をクリックします。  
サービスコネクタの Informatica Marketplace ページが開きます。
5. アカウントを作成し、Informatica Marketplace にログインします。
6. 左側のパネルの **[製品カテゴリ]** リストから、**[Cloud アプリケーション統合]** をクリックします。  
使用可能なサービスコネクタを示すページが表示されます。  
次の図に、一部のサンプルサービスコネクタを示します。

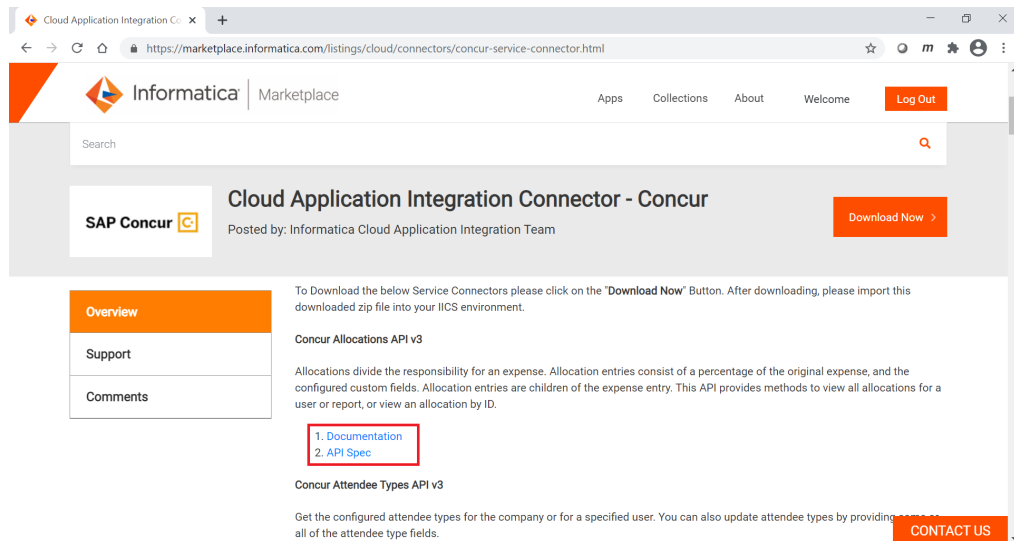


7. **【検索】** ボックスに、ダウンロードするサービスコネクタの名前を入力します。また、**【価格】** リストと**【タイプ】** リストを使用して、サービスコネクタをフィルタリングすることもできます。

8. ダウンロードするサービスコネクタの名前をクリックします。

サービスコネクタの詳細が表示されます。

サービスコネクタに特定の API ベースのサービスコネクタが含まれている場合は、次の図に示すように、すべてのコネクタの詳細と、API ドキュメントおよび API 仕様にアクセスするためのリンクが表示されます。



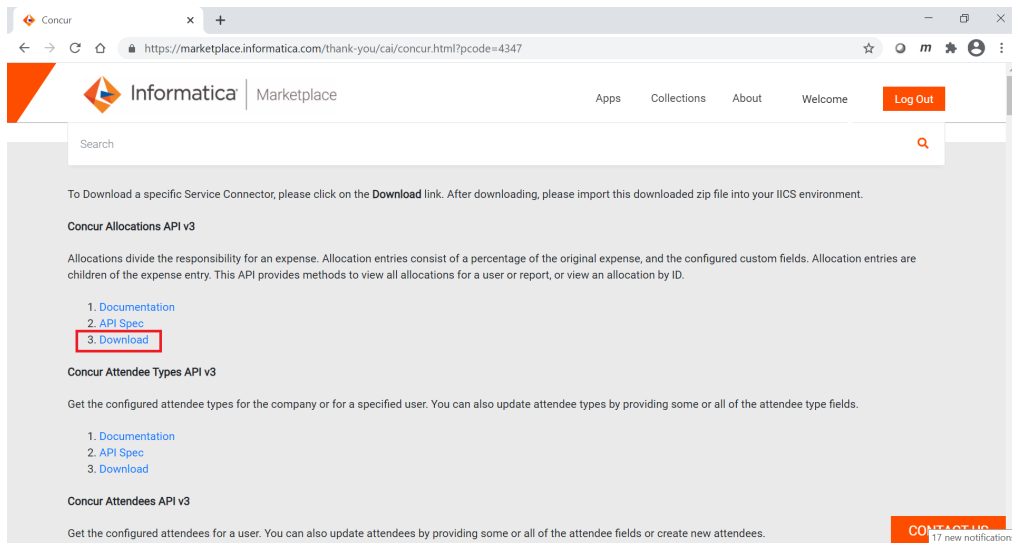
9. **【すぐにダウンロード】** をクリックします。

ユーザーの詳細の入力を求めるフォームが表示されます。

10. フォームに入力し、利用規約に同意して、フォームを送信します。

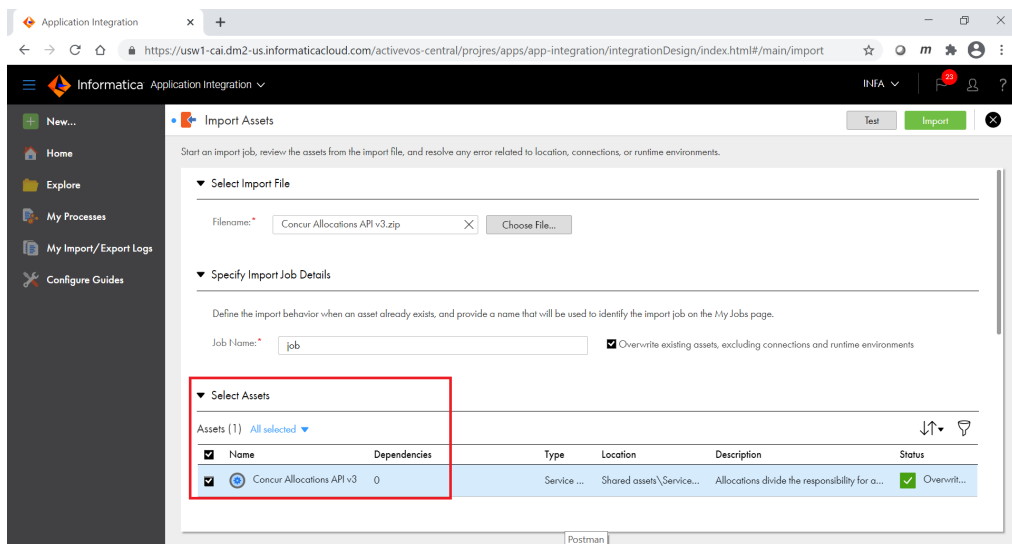
サービスコネクタがダウンロードされます。

サービスコネクタに特定の API ベースのサービスコネクタが含まれている場合は、次の図に示すように、特定のコネクタをダウンロードすることができます。



11. Informatica Intelligent Cloud Services にログインし、**[アプリケーション統合]** をクリックします。
12. **[参照]** ページで **[インポート]** をクリックして、Informatica Marketplace からダウンロードしたサービスコネクタの.zip ファイルを選択します。

次の図のように、**[アセットの選択]** セクションにサービスコネクタが表示されます。



13. **[インポート]** をクリックして **[マイインポート/エクスポートログ]** ページを開き、インポートのステータスを確認します。

インポートが完了したら、組織内の次の場所でサービスコネクタを利用できます。  
Shared assets\Service Connectors

# データアクセスサービスコネクタの作成

Secure Agent でデータアクセスサービスコネクタを作成し、データベースから直接データにアクセスできます。

データアクセスサービスコネクタを使用して、次のデータベースに接続できます。

- IBM DB2
- Microsoft SQL Server
- MySQL
- Oracle
- PostgreSQL

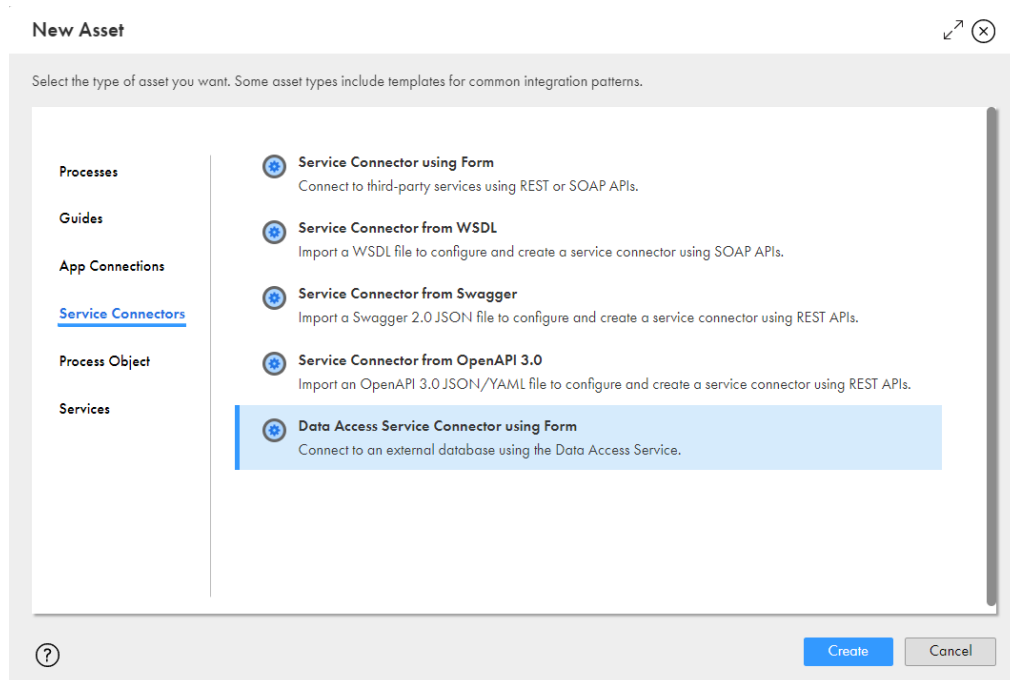
データアクセスサービスコネクタは、データベースと直接対話するためのプロパティを定義し、変数を指定することで作成できます。また、データアクセスサービスコネクタと関連付けられたアクションを定義することもできます。

データアクセスサービスコネクタを作成する前に、必要な JDBC JAR ファイルを次の場所にコピーし、Secure Agent を再起動する必要があります。

<Secure Agent のインストールディレクトリ>/apps/process-engine/ext

データアクセスサービスコネクタを作成するには、次の手順を実行します。

1. アプリケーションの統合で、**[新規]** > **[サービスコネクタ]** をクリックします。
2. **[新しいアセット]** ダイアログボックスで、**[フォームを使用したデータアクセスサービスコネクタ]** をクリックし、**[作成]** をクリックします。



## プロパティの定義

データアクセスサービスコネクタを作成してプロパティを定義するには、次の手順を実行します。

1. アプリケーションの統合で、**[新規] > [サービスコネクタ] > [フォームを使用したデータアクセスサービスコネクタ] > [作成]** をクリックします。

データアクセスサービスコネクタエディタが表示されます。

次の図は、データアクセスサービスコネクタエディタを示しています。

The screenshot shows the 'Definition' tab of the connector editor. The 'Name' field is 'DASConnectorForOData' with a note '(Unpublish connector to edit name)'. The 'Location' field is 'dupadhyay/ODataForDAS' with a 'Browse' button. The 'Description' field is empty. There are checkboxes for 'Agent Only' (unchecked) and 'Use OData' (checked). At the top right, there are 'Save' and 'Publish' buttons and a menu icon.

2. **[定義]** タブで、データアクセスサービスコネクタについて次の基本プロパティを定義します。

- **名前:** データアクセスサービスコネクタをプロセスで使用できるようにする名前。名前は英字または数字で始まる必要があり、英数字、マルチバイト文字、アンダースコア ( \_ )、およびハイフン ( - ) のみを含めることができます。128 文字を超える名前は使用できません。これは必須フィールドです。
- **場所:** データアクセスサービスコネクタを保存するプロジェクトまたはフォルダを指定します。これは必須フィールドです。
- **説明:** 必要に応じて、データアクセスサービスコネクタの説明を入力します。
- **エージェントのみ:** データアクセスサービスコネクタは Secure Agent でのみ実行できるため、これがデフォルトで選択されています。
- **OData の使用:** OData を使用する Web サービスからデータにアクセスするには、このオプションを選択します。データアクセスサービスコネクタの **[OData の使用]** オプションを有効にした後に、OData を有効にして、データアクセスサービス接続を使用するアプリケーション接続で、OData で許可されたユーザーとグループを指定できます。

アプリケーション接続に関連するプロパティの詳細については、「**デザイン**」を参照してください

アプリケーション接続をパブリッシュした後に、アプリケーション接続の **[プロパティの詳細]** ページで、OData サービス URL と OData Swagger URL を確認できます。

3. **[接続プロパティ]** セクションで、外部データベースに接続するための次の全般プロパティを定義します。

- **JDBC ドライバ:** JDBC ドライバの完全修飾 Java クラス名。  
データベースに基づいて、次のいずれかのドライバクラス名を指定できます。
  - **IBM DB2:** jdbc.db2.DB2Driver
  - **Microsoft SQL Server:** com.microsoft.sqlserver.jdbc.SQLServerDriver
  - **MySQL:** jdbc.mysql.MySQLDriver
  - **Oracle:** oracle.jdbc.OracleDriver
  - **PostgreSQL:** org.postgresql.Driverこれは必須フィールドです。
- **JDBC URL:** 外部データベースに接続するための接続 URL。  
データベースに基づいて、次のいずれかの URL を指定できます。
  - **IBM DB2:** jdbc:db2://<server>:<port>/<database>



- **Microsoft SQL Server:** jdbc:sqlserver://<host>\<instance>:<port>;databaseName=<database>
- **MySQL:** jdbc:mysql://<host>:<port>/<database>
- **Oracle:** jdbc:oracle:thin:@//<host>:<port>/<service>
- **PostgreSQL:** jdbc:postgresql://<host>:<port>/<database>

これは必須フィールドです。

- **ユーザー名:** データベースに接続するためのユーザー名。これは必須フィールドです。
  - **パスワード:** データベースに接続するためのパスワード。これは必須フィールドです。このフィールドに入力された値は、デフォルトで暗号化されます。開発者コンソールには、コネクタを保存するときに入力したパスワードが表示されます。コネクタを保存して閉じ、再度開くと、パスワードが暗号化されて表示されます。
  - **スキーマ:** メタデータに含めるテーブルまたは除外するテーブルを含むスキーマの名前を入力します。適切な結果を表示するには、データアクセスサービスコネクタを使用するアプリ接続にスキーマ名を入力する必要もあります。
  - **テーブルを含める:** メタデータに含めるテーブルの名前を入力します。テーブルのリストを含めるには、カンマを使用して複数のテーブル名を区切ります。パターンマッチングの場合は、「.\*」を使用することもできます。適切な結果を表示するには、データアクセスサービスコネクタを使用するアプリ接続に含まれるテーブル名を指定する必要もあります。
  - **テーブルを除外:** メタデータから除外するテーブルの名前を入力します。テーブルのリストを除外するには、カンマを使用して複数のテーブル名を区切ります。パターンマッチングの場合は、「.\*」を使用することもできます。適切な結果を表示するには、データアクセスサービスコネクタを使用するアプリ接続から除外されるテーブル名を指定する必要もあります。
4. 必要に応じて、**[接続プロパティ]** セクションで、次の詳細プロパティを定義します。
- **SQL インジェクションを確認:** 悪意のある入力パラメータを通じて発生する可能性のある SQL インジェクション攻撃を防止または軽減するには、このフィールドを **[はい]** に設定します。デフォルトは **[いいえ]** です。データアクセスサービスコネクタを使用するアプリ接続でもこのフィールドを設定する必要があります。
  - **初期接続プールサイズ:** プールの開始時に作成される接続の初期数。デフォルトは 5 です。
  - **最大アクティブ接続数:** プールから同時に割り当てることができるアクティブな接続の最大数。負の値の場合、制限はありません。デフォルトは 10 です。
  - **最小アイドル接続数:** 他の接続を作成せずにプール内でアイドル状態を維持できる接続の最小数。値 0 は何も作成されないことを意味します。デフォルトは 1 です。
  - **最大アイドル接続数:** 他の接続を解放せずにプール内でアイドル状態を維持できる接続の最大数。負の値の場合、制限はありません。デフォルトは 5 です。
  - **最大待機時間:** 使用可能な接続がない場合に、プールが接続を待機する最大ミリ秒数。値-1 は、無限に待機することを意味します。
  - **トランザクション分離レベル:** プールによって作成された接続のデフォルト状態で、次のいずれかの状態です。
    - 0 - なし。
    - 1 - コミットされていない読み取り
    - 2 - コミットされた読み取り
    - 4 - 反復可能な読み取り
    - 8 - シリアライズ可能
  - **最小イビクシオン可能アイドル時間:** オブジェクトが、削除される前にプール内でアイドル状態でいられる最小時間（イビクタがある場合）。

- **イビクションごとの数値テスト:** アイドルオブジェクトイビクタスレッドの各実行中にテストするオブジェクトの数（イビクタがある場合）。
- **借用時テスト:** 選択した場合、オブジェクトはプールから借用される前に検証されます。オブジェクトを検証できない場合はプールから破棄され、別の借用が新たに試みられます。デフォルトは false です。
- **リターン時テスト:** 選択した場合、借用したオブジェクトはプールに返される前に検証されます。デフォルトは false です。
- **アイドル中のテスト:** 選択した場合、借用したオブジェクトは、アイドルオブジェクトイビクタ（ある場合）によって検証されます。オブジェクトを検証できない場合はプールから破棄されます。デフォルトは false です。
- **検証クエリ:** 呼び出し元に返す前にプールの接続を検証する SQL クエリ。指定する場合、クエリは 1 つ以上の行を返す SQL SELECT ステートメントであることが必要です。

## 変数の指定

データアクセスサービスコネクタを設定するには、変数を指定して出力フィールドを定義することが必要になる場合があります。

### ビルトイン変数

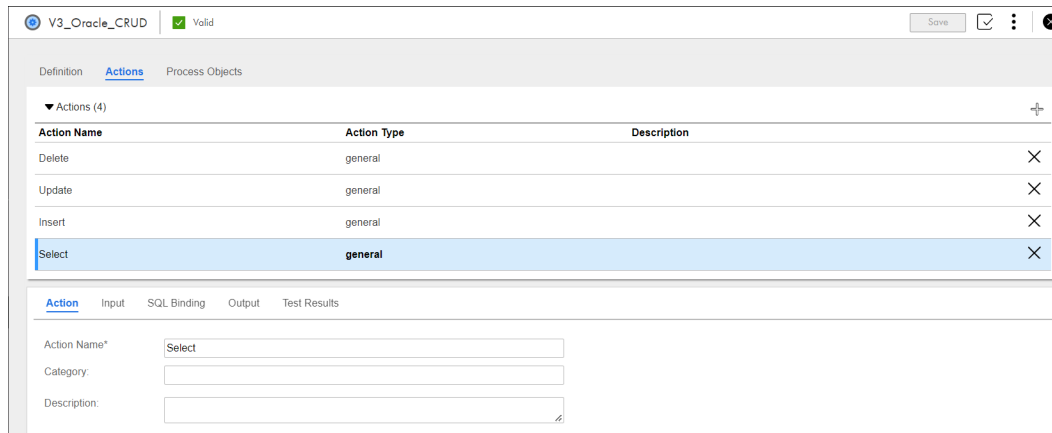
次の変数は、XQuery 式を使用してデータアクセスサービスコネクタで参照できます。

変数	変数型	説明
\$VariableName	すべての接続プロパティ	入力フィールドと出力フィールドは、この形式を使用して指定できます。
\$Response	出力フィールドマッピング	応答 XML データが含まれます。これは、データアクセスサービスコネクタの <b>【テスト結果】</b> タブに表示されます。

## アクションの定義

**【アクション】** タブでは、データアクセスサービスコネクタと関連付けられた 1 つ以上のアクションを作成して記述できます。

次の図は、**【アクション】** タブを示しています。



編集する行をクリックするか、[+] をクリックして新しい行を追加し、[アクション] タブで次の情報を入力します。

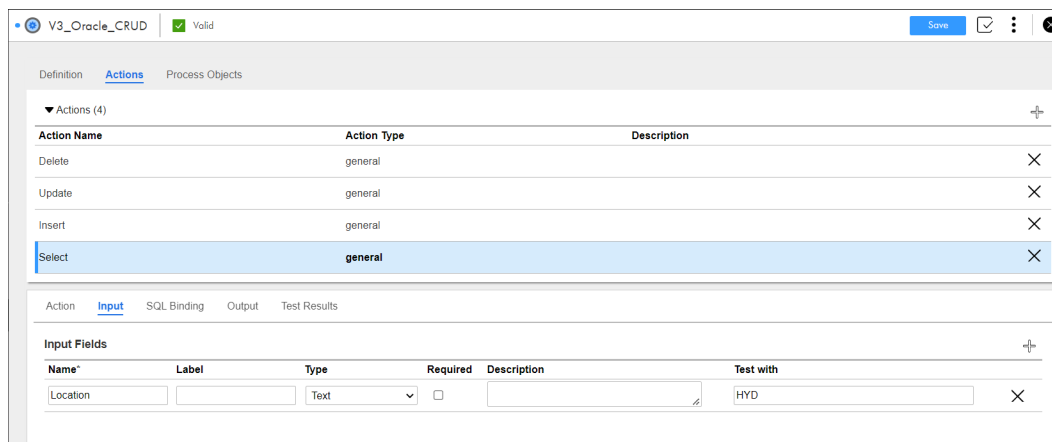
- **アクション名:** データアクセスサービスコネクタと接続でこのアクションを参照するときに表示する名前を入力します。アクション名にスペースまたは特殊文字は使用しないでください。これは必須フィールドです。
- **カテゴリ:** データアクセスサービスコネクタが多数あり、それぞれに複数のアクションがある場合は、カテゴリを作成して、ユーザーがプロセス内を移動しやすくなるようにすることができます。
- **説明:** このアクションの説明または付記を入力します。

各アクションについて、[入力]、[SQL バインディング]、[出力]、[テスト結果] の各タブに追加の詳細を指定します。

## [入力] タブ

[入力] タブを使用して、データの送信先サービスに固有の入力データ項目を定義します。

次の図は、[入力] タブを示しています。



各項目に対し、次のプロパティを入力します。

- **名前:** 必須。入力データ項目の名前。  
**注:** 入力フィールド名にスペース文字を使用し、その入力フィールドを [SQL バインディング] タブの SQL クエリで使用すると、エラーが表示される場合があります。
- **ラベル:** プロセス内に表示する項目の名前。指定しない場合は、名前が表示されます。

- **タイプ:** リストから項目のデータ型を選択します。[参照] または [オブジェクトリスト] 型を選択した場合、プロセスオブジェクトも選択できます。
  - **必須:** このパラメータに値を設定する必要がある場合に選択します。
  - **説明:** パラメータの説明を入力します。
  - **テストに使用する値:** Process Designer がアクションをテストするために使用する値を入力します。入力が添付ファイルの場合、サンプルの添付ファイルをアップロードすることもできます。
- [+] アイコンを使用し、新しい項目を追加します。[X] をクリックして現在の行を削除します。

## [SQL バインディング] タブ

データアクセスサービスコネクタの各アクションについて、[SQL バインディング] タブを使用して、データソースからデータにアクセスするための複数の SQL クエリを指定します。SQL エントリが作成されると、タブには、SQL 名、SQL テキスト、式のカラムの大文字と小文字、クエリタイプ、SQL クエリによって返される最大行数、最大待機時間 (秒) などの詳細が表示されます。

複数のクエリを追加すると、テーブルにクエリがリストされている順序で文が実行されます。矢印ボタンを使用してクエリを上下に移動できます。SQL クエリを削除することもできます。

SQL エントリの行を追加するには、[SQL バインディング] タブの [+] をクリックします。各 SQL クエリの次の詳細を入力します。

### SQL 名前

SQL クエリの名前。

### SQL テキスト

少なくとも 1 行を返す有効な SQL 文。

### カラムの大文字と小文字

式内の文字を小文字または大文字に変換する必要があるか、または変更しないままにするかを指定します。デフォルトは [小文字] です。

### タイプ

次のいずれかの値を選択します。

- **データの操作。** 情報をクエリし、それにロジックを適用してまったく異なるデータセットを生成するには、この値を選択します。
- **ストアドプロシージャ。** 保存して再利用できる準備済みの SQL コードを使用するには、この値を選択します。

### 結果セットあり

将来の使用のために予約済み。

### 最大行数

SQL クエリによって返され、SQL 応答ペイロードに表示できる最大行数。最大行数は 10,000 行以下にする必要があります。

### 最大待機時間(秒)

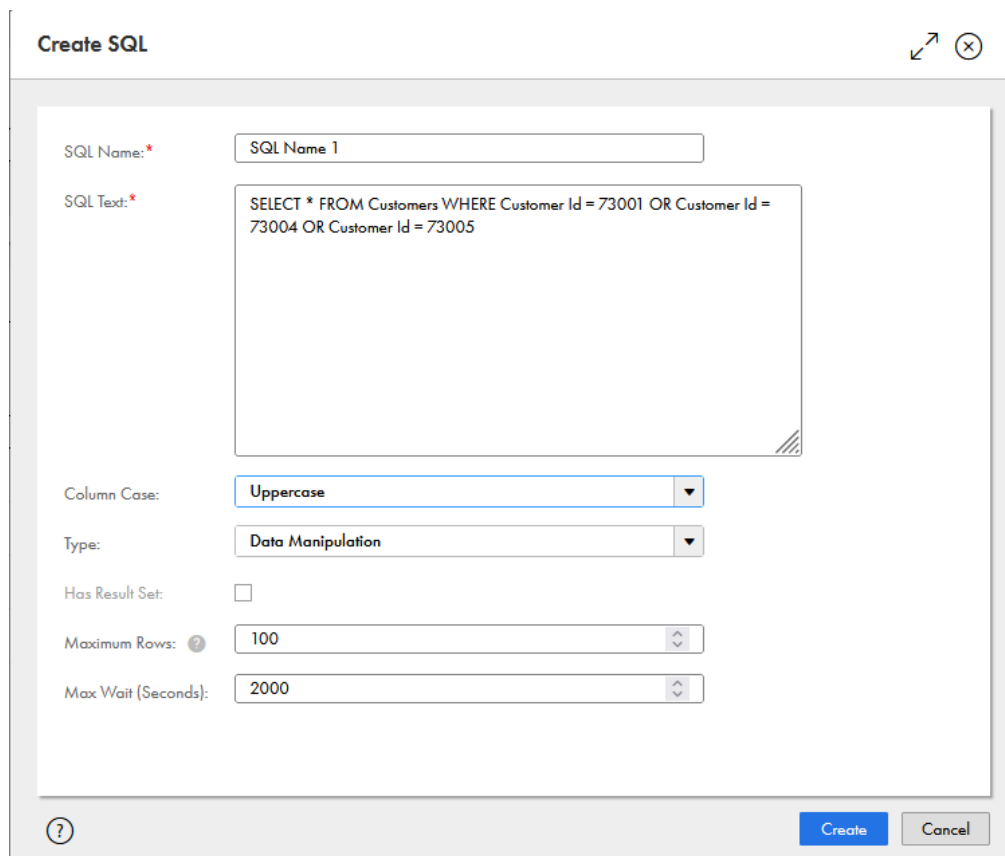
データベースクライアントがセッションをタイムアウトするまでにデータベースからの応答を待機できる最大時間 (秒)。

**注:** アプリケーション統合は、すべての SQL クエリを実行した後にコミットを実行しません。SQL クエリが失敗した場合のロールバックを支援するために、コミットはすべての SQL クエリが実行された後の最後のみ行われます。一部のデータベースでは、CREATE TABLE、DROP TABLE などの DDL 文に対して暗黙的なコミットが発生します。これらの実行はロールバックできません。

## 複数の SQL クエリの設定

SQL クエリを追加および編集するには、次の手順を実行します。

1. **【SQL バインディング】** タブで、**[+]** アイコンをクリックして SQL エントリを追加します。  
SQL エントリが **【SQL バインディング】** タブのテーブルに追加されます。
2. 編集する必要がある SQL クエリに対応した **【編集】** アイコンをクリックします。  
**【SQL の作成】** ページが表示されます。次の図は、複数の SQL クエリを設定できる **【SQL の作成】** ページを示しています。



3. **【SQL の作成】** ページで、SQL クエリのプロパティを設定します。
4. **【作成】** をクリックします。

## SQL クエリの応答形式

データアクセスサービスコネクタでは、単一および複数の SQL クエリに対する XML 応答は、multiDataAccessResponse タグから始まります。

次のスニペット例は、プロセスまたはデータアクセスサービスの出力としての形式を示しています。

```
<multiDataAccessResponse xmlns:aetgt="http://schemas.informatica.com/socrates/data-services/2014/05/business-connector-model.xsd" xmlns:bconn="http://schemas.informatica.com/socrates/data-services/2014/05/business-connector-model.xsd">
  <dataAccessResponse>
    <result>
      <statementId>SQL Name 1</statementId>
      <row>
```

```

        <updatedRows>0</updatedRows>
      </row>
    </result>
  </dataAccessResponse>
  <dataAccessResponse>
    <result>
      <statementId>SQL Name 2</statementId>
      <row>
        <updatedRows>1</updatedRows>
      </row>
    </result>
  </dataAccessResponse>
</multiDataAccessResponse>

```

次のスニペット例は、応答ペイロードとしての XML 形式を示しています。

```

<multiDataAccessResponse>
  <dataAccessResponse>
    <result statementId="SQL Name 1">
      <row>
        <updatedRows>0</updatedRows>
      </row>
    </result>
  </dataAccessResponse>
  <dataAccessResponse>
    <result statementId="SQL Name 2">
      <row>
        <updatedRows>1</updatedRows>
      </row>
    </result>
  </dataAccessResponse>
</multiDataAccessResponse>

```

## [出力] タブ

**[出力]** タブを使用して、データアクセスサービスコネクタがデータベースから返されたデータをどのように解析するかを定義します。

次の図は、**[出力]** タブを示しています。

The screenshot shows the Oracle APEX interface for a service connector. At the top, there's a breadcrumb 'V3\_Oracle\_CRUD' and a 'Valid' status indicator. Below that, there are tabs for 'Definition', 'Actions', and 'Process Objects'. The 'Actions' tab is active, showing a list of four actions: Delete, Update, Insert, and Select. The 'Select' action is selected and highlighted. Below the actions list, there are tabs for 'Action', 'Input', 'SQL Binding', 'Output', and 'Test Results'. The 'Output' tab is active, showing a table of 'Output Fields'. The table has four columns: 'Name\*', 'Type', 'Description', and 'Get From'. There are three rows of output fields:

Name*	Type	Description	Get From
Department number	Text		Column deptno
Department Name	Text		Column deptname
Location	Text		Column loc

各出力データ項目について、次のプロパティを指定します。

- **名前:** 返された値が配置されるフィールドの名前。
- **タイプ:** フィールドに書き込まれる値のデータ型。タイプが [オブジェクトリスト] または [参照] である場合、Process Designer はプロセスオブジェクトのリストを表示し、[プロセスオブジェクト] タブ内で定義されたオブジェクトのいずれかを選択できるようにします。
- **説明:** 出力フィールドの説明を入力します。
- **取得元:** 次のいずれかのオプションを選択します。
  - **カラム:** 出力フィールド内に配置されるカラム名（データベースから返される出力内で使用されるデータベーステーブル内のカラム名）を入力します。
  - **式:** データベースから返される出力を解析するための式を記述します。[f(x)] をクリックして、式を入力するための式エディタを開きます。
  - **結果セット:** データベースからの応答の完全な内容出力フィールドに割り当てます。
  - **添付ファイルとしてのすべての応答:** 応答全体を添付ファイルとして処理します。
  - **添付ファイル:** 複数の添付ファイルを処理して、添付ファイルの全リストを選択した変数（ペイロードとして使用されるパートを除く）に渡します。

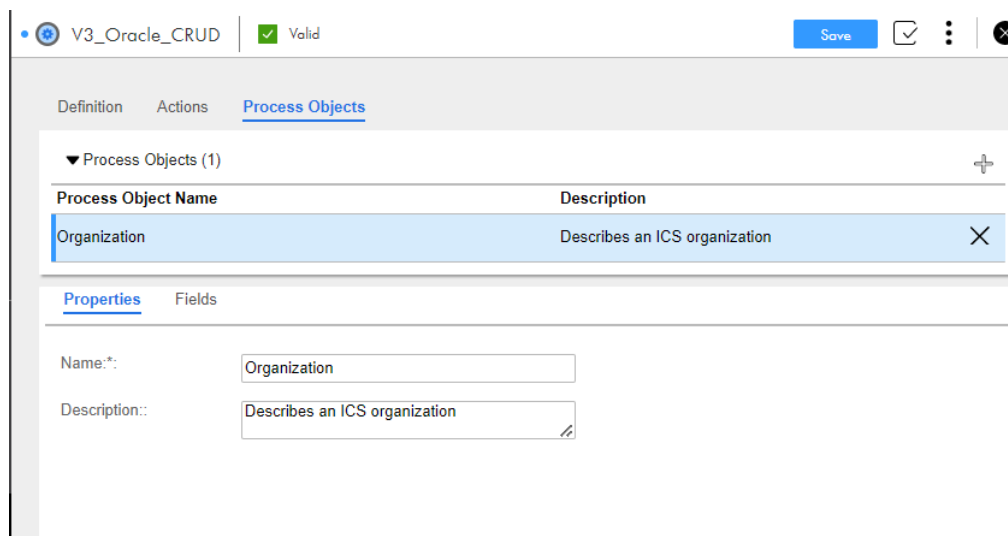
## プロセスオブジェクトの作成

[プロセスオブジェクト] タブでは、データアクセスサービスコネクタの1つ以上のプロセスオブジェクトを定義して、データをグループ化し、構造化オブジェクトを作成できます。データアクセスサービスコネクタで定義されたプロセスオブジェクトは、データアクセスサービスコネクタを使用するプロセスで利用可能になります。例えば、データベースが名前、住所、電話番号などの人口統計情報を返す場合、この情報を格納する単一の人口統計的プロセスオブジェクトを作成できます。

[アクション/出力] タブを使用して、データベースから返された1つ以上のデータ要素と各プロセスオブジェクトを関連付けることができます。

データアクセスサービスコネクタのプロセスオブジェクトを作成するには、次の手順を実行します。

1. データアクセスサービスコネクタを作成するか、既存のコネクタを開きます。
2. [プロセスオブジェクト] タブをクリックします。
3. [+] をクリックして新しいプロセスオブジェクトを追加するか、既存のプロセスオブジェクトをリストから選択します。各プロセスオブジェクトは、タブ内に行項目として表示されます。次の図は、[プロセスオブジェクト] タブを示しています。



4. **【プロパティ】** タブで、各プロセスオブジェクトに対し、次の詳細を指定します。
  - **名前:** プロセスオブジェクトを識別する名前を入力します。この名前は、プロセスオブジェクトを選択できるリストに表示されます。これは必須フィールドです。
  - **説明:** プロセスオブジェクトの説明を入力します。
5. **【フィールド】** タブで、各プロセスオブジェクトに対し、次の詳細を指定します。
  - **フィールド名:** プロセスオブジェクト内の各フィールドの名前を入力します。
  - **タイプ:** テキスト、整数、日付/時刻などの組み込みデータ型のいずれかを使用して、フィールドに書き込まれる値のデータ型を選択します。[オブジェクトリスト] 型の場合、プロセスオブジェクトも選択します。
  - **必須:** このチェックボックスを選択すると、フィールドが必須フィールドとして設定されます。必須フィールドは、要求ペイロードに含まれている必要があります。値を含めることも、空にすることも、NULL にすることもできます。
  - **NULL 可能:** フィールドが NULL 値を受け入れないようにする場合は、このチェックボックスをクリアします。デフォルトでは、**【NULL 可能】** チェックボックスが選択されています。

## データアクセスサービスコネクタのテスト

データアクセスサービスコネクタエディタを使用するときは、**【テスト】** をクリックしてデータベースに要求を送信し、応答データを **【テスト結果】** タブに表示します。

- **プロセスオブジェクトの生成:** クリックすると、データアクセスサービスコネクタ用に定義したプロセスオブジェクトが表示されます。
- **結果:** テストのステータスを表示します。
- **出力および値:** **【出力】** タブで定義されたフィールドに割り当てられたデータベースから返されたデータを表示します。
- **応答ペイロード:** データベースから Process Designer に送信されたペイロード（データベースから送信された完全な応答ではない）を表示します。
- **SQL 要求:** データアクセスサービスコネクタからデータベースに送信されたデータを表示します。このタブの左側のペインには SQL 名のリストが表示され、右側のペインには対応するデータが表示されます。
- **SQL 応答:** データベースからデータアクセスサービスコネクタに返されたすべてのデータを表示します（応答ペイロードは、返されたデータの一部にすぎない）。このタブの左側のペインには SQL 名のリストが表示され、右側のペインには対応するデータが表示されます。



応答に1つ以上の添付ファイルが含まれている場合は、それらの添付ファイルをダウンロードすることもできます。

**注:** [応答ペイロード] および [SQL 応答] セクションには最大 10,000 件のレコードが表示されます。[SQL バインディング] タブで、10,000 を超える値を制限に指定することはできません。

次の図は、[テスト結果] タブを示しています。

Action Name	Action Type	Description
insertInto	General	
Select	General	

Test Server: INWPF3M1V4D-AAD [Test]

Result:

Output Fields [Generate Process Objects]

Response Payload	SQL Request	SQL Response
SQL Name 1	1 <dataAccessResponse>	
	2 <dataAccessResponse><result><statementId>SQL Name 1</statementId><row><reg_code>3</reg_code><lastname>xyz1</lastname><firstname>abc1</firstname></row><reg_code>1</reg_code><lastname>xyz</lastname>	
	3 <dataAccessResponse><result><statementId>SQL Name 2</statementId><row><eid>10</eid><ename>keet1</ename><edpt>test</edpt></row><row><eid>20</eid>	
	4 <dataAccessResponse><result><statementId>SQL Name 1</statementId><row><reg_code>3</reg_code>	
	5 <dataAccessResponse><result><statementId>SQL Name 1</statementId><row><reg_code>3</reg_code>	
	6 <dataAccessResponse><result><statementId>SQL Name 1</statementId><row><reg_code>3</reg_code>	
	7 <dataAccessResponse><result><statementId>SQL Name 1</statementId><row><reg_code>3</reg_code>	
	8 <dataAccessResponse><result><statementId>SQL Name 1</statementId><row><reg_code>3</reg_code>	

## プロセスオブジェクトの生成

通常プロセスオブジェクトは、特定の要素セットに結び付けられます。場合によっては、多数の同一のオブジェクトを持ち、それぞれが1つの要素セットに適用されることがあります。例えば、返されるデータを定義するとき、返されるデータのサブセットを定義することのみを目的とするプロセスオブジェクトを作成します。これらのオブジェクトは再利用不可能であり、単一のプロセスオブジェクトフィールドのみが使用できます。オブジェクト名は、フィールドの名前でもあります。

再利用可能なオブジェクトを作成することもできます。例えば、NetSuite の refType 要素には、name と internalID という2つのフィールドがあります。これら2つのフィールドをそれぞれが含む多数のオブジェクトを作成するのではなく、1つの再利用可能なプロセスオブジェクトを作成できます。

1. [プロセスオブジェクトの生成] をクリックして、データアクセスサービスコネクタ用に定義したプロセスオブジェクトを表示します。
2. 再利用可能にするプロセスオブジェクトを選択し、[次へ] をクリックします。Process Designer に、生成されたプロセスオブジェクトの一覧が表示されます。
3. [完了] をクリックします。

## 応答ペイロード内のメタデータパラメータ

特定のアプリケーション接続のメタデータの詳細を表示するには、メタデータパラメータを使用します。出力には、データと一緒に保存されているデータまたはデータセットに関するコンテキスト情報が含まれます。

アプリケーション接続のメタデータの詳細を表示するには、次の URI を使用します。

`https://<Informatica Intelligent Cloud Services URL>/active-bpel/odata/v4/<app connection name>/$metadata`

次のスニペット例は、応答ペイロードの形式を示しています。

```
▼<Annotation Term="Informatica.OData.Screenflow.V1.Required">
  <Bool>true</Bool>
</Annotation>
</Property>
<Property Name="psg_container" Type="Edm.Decimal" Precision="19"/>
<Property Name="pog_name" Type="Edm.String" MaxLength="765"/>
</EntityType>
▼<EntityType Name="StudentODataTest">
  ▼<Key>
    <PropertyRef Name="studentid"/>
  </Key>
  ▼<Property Name="studentid" Type="Edm.Int64" Nullable="false">
    ▼<Annotation Term="Informatica.OData.Screenflow.V1.Required">
      <Bool>true</Bool>
    </Annotation>
    ▼<Annotation Term="Informatica.OData.Screenflow.V1.ReadOnly">
      <Bool>true</Bool>
    </Annotation>
  </Property>
  ▼<Property Name="firstname" Type="Edm.String" MaxLength="255">
    ▼<Annotation Term="Informatica.OData.Screenflow.V1.Required">
      <Bool>true</Bool>
    </Annotation>
  </Property>
  ▼<Property Name="lastname" Type="Edm.String" MaxLength="255">
    ▼<Annotation Term="Informatica.OData.Screenflow.V1.Required">
      <Bool>true</Bool>
    </Annotation>
  </Property>
  <Property Name="sectionid" Type="Edm.Int64" MaxLength="10"/>
  <Property Name="subjectid" Type="Edm.Int64" MaxLength="10"/>
  ▼<NavigationProperty Name="SectionODataTest" Type="Informatica.OData.AppConnectionTestSqlServer1.Models.SectionODataTest">
    <ReferentialConstraint Property="sectionid" ReferencedProperty="sectionid"/>
  </NavigationProperty>
  ▼<NavigationProperty Name="SubjectODataTest1" Type="Informatica.OData.AppConnectionTestSqlServer1.Models.SubjectODataTest1">
    <ReferentialConstraint Property="subjectid" ReferencedProperty="subjectid"/>
  </NavigationProperty>
</EntityType>
```

## 応答ペイロード内の OData 拡張パラメータ

アプリケーション接続をパブリッシュした後、OData 拡張パラメータを使用して、データに対してネストされたクエリを実行できます。サービス呼び出しでは、ナビゲーションプロパティの 1 レベルまでをクエリできません。

OData 拡張パラメータを含む次の URI を使用します。

`https://<Informatica Intelligent Cloud Services URL>/active-bpel/odata/v4/<app connection name>/<table name>?$expand=*`

次のスニペット例は、拡張クエリの出力を示しています。

```

3      "value": [
4        {
5          "studentid": 1,
6          "firstname": "FirstName1",
7          "lastname": "FisrtName2",
8          "sectionid": 11,
9          "subjectid": 111,
10         "sectionodata": {
11           "sectionid": 11,
12           "numberofstudents": 10
13         }
14       },
15       {
16         "studentid": 2,
17         "firstname": "FirstName2",
18         "lastname": "FisrtName2",
19         "sectionid": 22,
20         "subjectid": 222,
21         "sectionodata": {
22           "sectionid": 22,
23           "numberofstudents": 20
24       }
25     }
26   ]

```

## サービスコネクタの検証

サービスコネクタまたはデータアクセスサービスコネクタを作成するときに、[検証] パネルを使用して、検証エラーの有無を調べたりエラーの詳細を確認したりできます。

1. 次の図に示すように、ツールバーの [検証] アイコンをクリックします。



右ペインに [検証] パネルが表示されます。[検証] パネルでは、発生したタブに基づいてエラーが分類されています。

2. 次の図に示すように、タブ名に対応する矢印をクリックします。



そのタブで発生したエラーのリストが表示されます。

3. 次の図に示すように、エラー行をクリックします。

The screenshot shows the ServiceConnector interface. At the top, there are tabs for 'Definition', 'Actions', and 'Process Objects'. The 'Actions' tab is active, showing a table with one row: 'Action Name' (empty) and 'Action Type' (general). Below the table, the 'Action' configuration panel is visible. The 'Action Name\*' field is empty and highlighted with a red box. Below it, the error message 'The Name field is required.' is displayed. To the right, the 'Validation' panel shows a list of errors. The first error is highlighted in red: 'Row 1 > Action: The Name field is required.' There are also other errors listed, such as 'Row 1 > Binding: URL Should not be empty.' and 'Row 1 > Properties: The Name field is required.'

問題のあるフィールドにカーソルが移動します。問題に対処してエラーを解決します。

エラーを修正すると、**[検証]** パネルのエラーリストが更新されます。

## サービスコネクタの保存およびパブリッシュ

サービスコネクタまたはデータアクセスサービスコネクタを作成したら、サービスコネクタを保存してパブリッシュします。

複数のサービスコネクタをまとめてパブリッシュすることもできます。詳細については、「呼び出し」を参照してください。

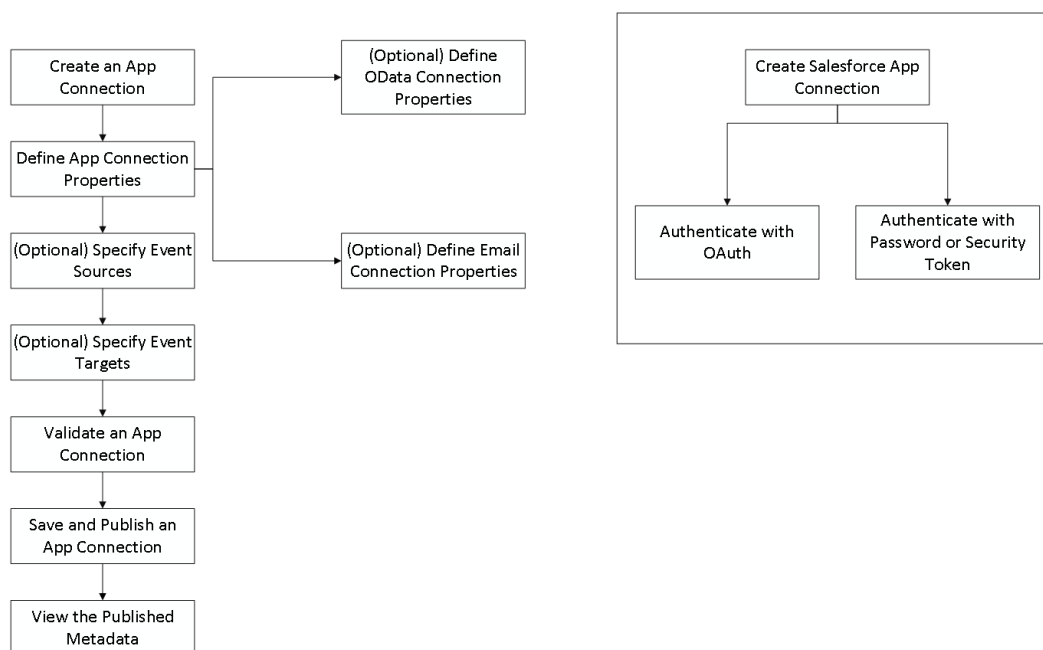
サービスコネクタデザイナーページで、**[保存]** をクリックしてサービスコネクタを保存し、**[パブリッシュ]** をクリックしてサービスコネクタをパブリッシュします。

パブリッシュしたサービスコネクタを無効にするには、そのサービスコネクタをパブリッシュ解除する必要があります。必要な変更を加えた後、そのサービスコネクタを再びパブリッシュすると、それらの変更が反映されます。

## 第 7 章

# コネクタの使用

サービスコネクタを作成したら、プロセスで使用するアプリケーション接続を作成します。  
次の図に、アプリケーション接続を作成するために実行する必要があるステップを示します。

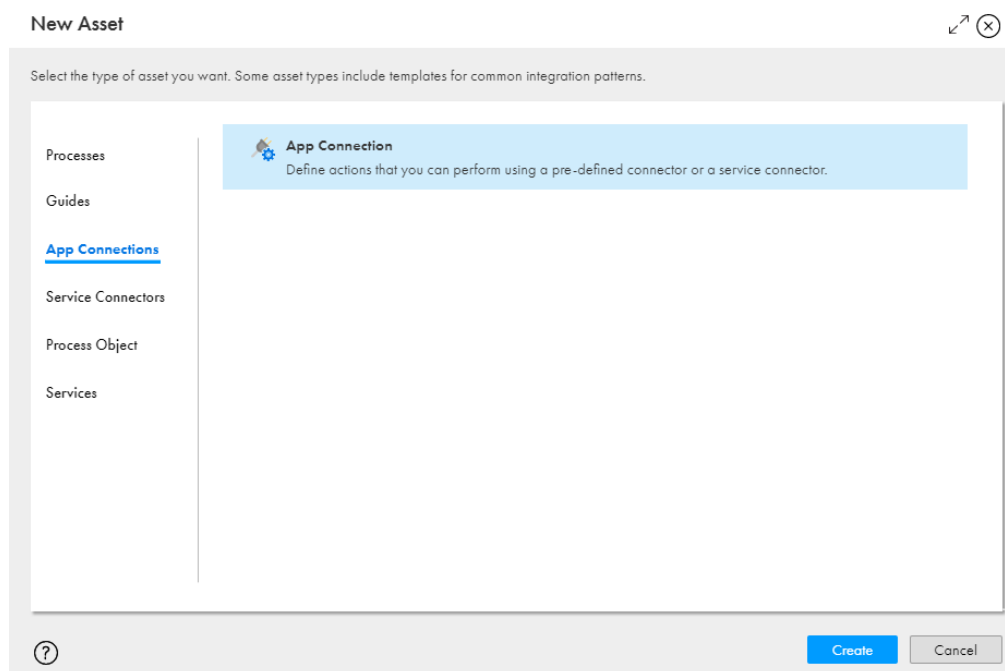


## アプリケーション接続の作成

アプリケーション接続を作成するには、以下の手順を実行します。

1. アプリケーションの統合で、**[新規]** をクリックします。

2. 次の図に示すように、[新しいアセット] ダイアログボックスで、[アプリケーション接続] > [アプリケーション接続] をクリックし、[作成] をクリックします。



## アプリケーション接続プロパティの定義

必要に応じて、各アプリケーション接続に対して説明的なプロパティを入力し、範囲を特定し、認証パラメータを設定します。

これらのオプションは、すべてのアプリケーション接続に表示されます。

- **接続タイプ:** このリストから、パブリッシュされたサービスコネクタ、ネイティブのツールキットコネクタ (JDBC など)、Web サービスコネクタ (Salesforce など) を選択します。接続を作成するとき、選択した接続タイプによって表示される接続プロパティが決定されます。

**注:** アプリケーション接続をパブリッシュすると、接続タイプは変更できなくなります。

- **名前:** プロセスおよび Designer 内でアプリケーション接続を特定する名前を入力します。名前は英字で始まる必要があり、英数字、マルチバイト文字、およびハイフン (-) のみを含めることができます。128 文字を超える名前は使用できません。これは必須フィールドです。

**重要:** Salesforce 内のプロセスのパブリッシュに使用するアプリケーション接続を定義する場合、[接続名] は Salesforce である必要があります。これは、プロセスが Salesforce 内のユーザーからアクセスできるようにするためです。1 つの Salesforce アプリケーション接続のみが、各アプリケーションの統合組織に対して許可されます。

- **説明:** アプリケーション接続を特定するための説明を入力します。
- **実行:** ファイアウォールの向こう側のサービスへのアクセスに使用するエージェントを選択するか、[クラウドサーバーまたは任意の Secure Agent] を選択します。このリストには、組織内で使用できるエージェントが表示されます。
- **接続テスト:** クラウドサーバーまたは [実行] フィールドに指定されたエージェント上で実行された任意の接続テストの結果を表示します。

- **OData 対応:** **【はい】** を選択して、OData フィードを有効にします。**【はい】** を選択する場合、設計時に接続にアクセス可能な許可されたユーザーまたは許可されたグループを指定する必要があります。OData エンドポイント URL へのクラウドアクセスを有効または無効にすることもできます。

アプリケーション接続で OData を有効にし、接続をバブリッシュした後、**【アクション】** > **【プロパティの詳細】** をクリックして、生成された OData サービス URL と OData Swagger URL を表示します。

OData サービスの URL は Informatica Cloud エンドポイント上にあります。使用される形式は次のとおりです。

`https://<Informatica Intelligent Cloud Services URL>/active-bpel/odata/v4/<OData_connection_name>`

OData Swagger URL は Informatica Cloud エンドポイント上にあります。使用される形式は次のとおりです。

`https://<Informatica Intelligent Cloud Services URL>/active-bpel/odata/v4/<connection_name>/$metadata?swagger`

データアクセスサービスコネクタで OData プロトコルを有効にする方法の詳細については、「**プロパティの定義**」を参照してください。

- **OData クラウドアクセス有効:** OData を有効にし、接続を Secure Agent マシンまたは Secure Agent グループで実行するように設定する場合、セキュリティの目的で OData エンドポイント URL へのクラウドアクセスを無効にすることができます。

- **OData で許可されたユーザー/グループ:**

実行時にこのアプリケーション接続へのアクセス権を持つユーザー。このフィールドには、複数のユーザーを入力できます。最初の値を指定したら、Enter キーを押すかカンマを入力してから次の値を指定します。

**OData で許可されたロール:**

実行時にこのアプリケーション接続へのアクセス権を持つロール。これらのロールに割り当てられたユーザーは、OData エンドポイント URL へのアクセス権限を持ちます。カスタムロールまたはシステム定義ロールを入力できます。このフィールドには、複数のロールを入力できます。

**重要:** カスタムロールを **【OData で許可されたロール】** フィールドに入力する場合、OData エンドポイント URL を起動するユーザーは、そのカスタムロールとサービスコンシューマロールを持っているか、またはそれらを持つグループに属している必要があります。

例:

自分の組織に、アプリケーション統合に対して有効な実行権限を持つカスタムロール、Testers があるとします。プロセスを作成し、**【許可されたロール】** フィールドに「Testers」を入力します。

ユーザーが OData エンドポイント URL を使用できるようにするには、次の条件が満たされる必要があります。

- 条件 1: ユーザーに、このアプリケーション接続が要求するロールである Testers ロールがある。
- 条件 2: ユーザーに、サービスコンシューマロールがある。アプリケーション統合は、任意のエンドポイント URL を起動するために、このロールを必要とします。

Testers ロールとサービスコンシューマロールを持つユーザーは、OData エンドポイント URL を起動できます。これは、条件 1 と条件 2 の両方が満たされているためです。

Testers ロールのみを持つユーザーは、OData エンドポイント URL を起動できません。これは、条件 1 が満たされているものの、条件 2 が満たされていないためです。

同様に、サービスコンシューマロールのみを持つユーザーは、OData エンドポイント URL を起動できません。これは、条件 2 が満たされているものの、条件 1 が満たされていないためです。

- **スキーマ:** メタデータに含めるテーブルまたは除外するテーブルを含むスキーマの名前を入力します。適切な結果を表示するには、データアクセスサービスコネクタを使用するアプリ接続にスキーマ名を入力する必要があります。
- **テーブルを含める:** メタデータに含めるテーブルの名前を入力します。テーブルのリストを含めるには、カンマを使用して複数のテーブル名を区切ります。パターンマッチングの場合は、「.\*」を使用することもで

きます。適切な結果を表示するには、データアクセスサービスコネクタを使用するアプリ接続に含まれるテーブル名を指定する必要があります。

- **テーブルを除外:** メタデータから除外するテーブルの名前を入力します。テーブルのリストを除外するには、カンマを使用して複数のテーブル名を区切ります。パターンマッチングの場合は、「.\*」を使用することもできます。適切な結果を表示するには、データアクセスサービスコネクタを使用するアプリ接続から除外されるテーブル名を指定する必要があります。

残りのプロパティは、コネクタまたは選択した接続タイプによって決定されます。プロパティ値を入力した後に、プロパティ行の外側の任意の場所をクリックするか、Enter キーを押して値を保存します。

例えば、次の図の Salesforce 接続タイプには、**[名前]**、**[タイプ]**、**[承認 URL]**、および **[トークン要求 URL]** が必要です。

The screenshot shows the configuration interface for a Salesforce connection. The 'Connection Details' section includes fields for Name (SalesforceAppConn), Location (Default), Description (description), Type (Salesforce), Run On (Cloud Server or any Secure Agent), Connection Test (Not Supported), OData-Enabled (No), Allowed Users for OData, and Allowed Groups for OData. The 'Authentication' section shows Authentication Type (OAuth) and a table of authentication-related properties.

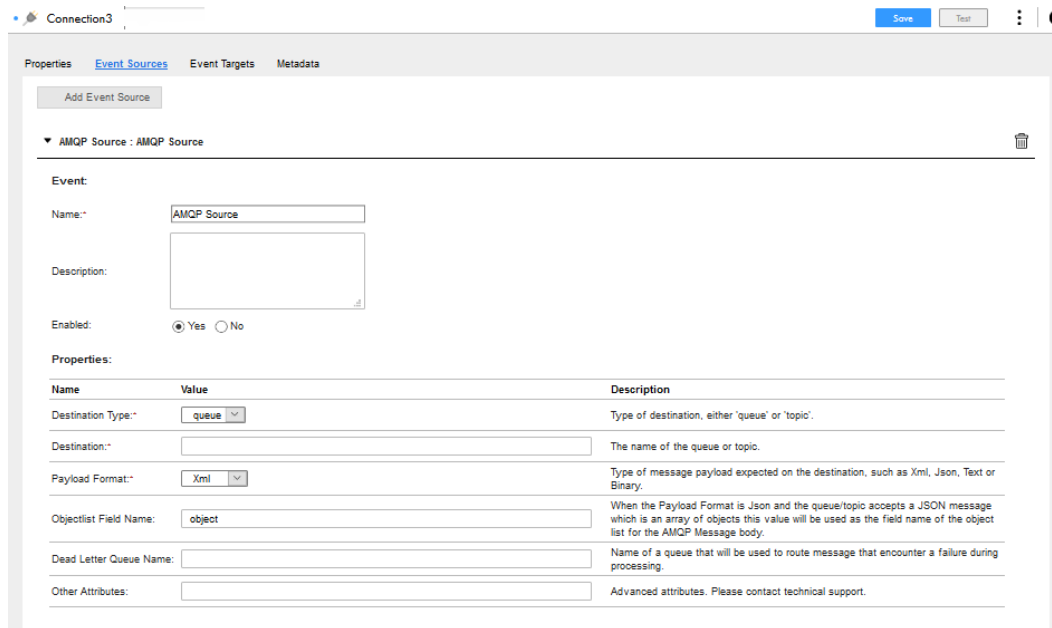
Name	Value	Description
Authorization URL	https://login.salesforce.com/services/oauth2/authorize	Enter the Salesforce OAuth authorization URL. Default value for production is https://login.salesforce.com/services/oauth2/authorize. For Sandbox, use https://test.salesforce.com/services/oauth2/authorize.
Token Request URL	https://login.salesforce.com/services/oauth2/token	Enter the OAuth token request URL. For production, use https://login.salesforce.com/services/oauth2/token. For sandbox, use https://test.salesforce.com/services/oauth2/token.
Session Duration	60	Enter the number of minutes to wait before refreshing the session. Default is 60 minutes.
Authorization Status	Not yet authorized.	Indicates the current status and the last time that authorization was completed.
Authorize Access	Authorize	Click to initiate the authorization workflow using OAuth.

## イベントソースの指定

イベントソースは、サービスコネクタに基づいて、いくつかのアプリケーション接続で使用できます。

AMQP リスナーサービスへのアクセスを提供する次の AMQP アプリケーション接続では、1 つ以上のイベントソースを追加できます。





まず、このアプリケーション接続に新しいイベントソースを追加するには、**【イベントソースの追加】** をクリックします。次に、各イベントについて次の情報を入力します。

- **名前:** 各イベントソースの一意の名前を入力します。名前は先頭の文字を英字にする必要があり、名前には英数字とハイフン (-) のみを含めることができます。これは必須フィールドです。
- **説明:** 必要に応じて、イベントソースの説明を入力します。
- **有効:** イベントソースをデプロイする準備ができるまで無効にするには、**【いいえ】** を選択します。これにより、複数のイベントソースを設定し、いつ有効にするかを決定する柔軟性が得られます。

設定が必要な接続プロパティは接続タイプによって異なります。このケースでは、AMQP 接続には **【宛先タイプ】**、**【宛先】**、および **【ペイロード形式】** フィールドを設定する必要があります。

プレビューオプションが有効になっていないか、接続タイプでサポートされていない場合、これを通知するメッセージがタブに表示されます。

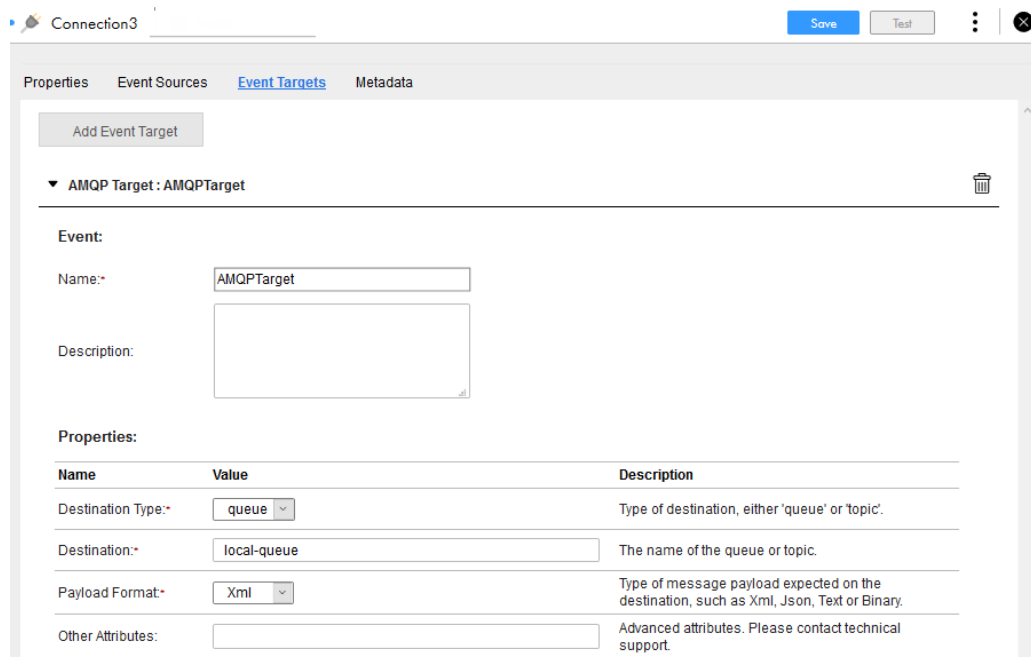
## イベントターゲットの指定

イベントソースと同様に、個別のタブの各アプリケーション接続に 1 つ以上のイベントターゲットを指定できます。

各コネクタに、異なるタイプのイベントターゲットを定義する場合があります。例えば、ファイルコネクタを使用して、定義する各イベントターゲットに対して、**ファイルライタ**または**区切りコンテンツライタファイルライタ**を選択できます。各イベントターゲットには、そのイベントターゲットタイプに固有の一連のプロパティが含まれます。

新しいイベントターゲットを作成する手順

1. **【イベントターゲットの追加】** をクリックします（該当する場合はターゲットタイプを選択します）。新しい行が追加されます。
2. 行を強調表示し、イベントの名前と説明を追加します。
3. 特定のコネクタのプロパティを編集します。



## アプリケーション接続の機密フィールド値の更新

更新リソースを使用して、アプリケーション接続の機密フィールドの値を更新できます。単一のアプリケーション接続または複数のアプリケーション接続のフィールドを更新できます。

更新リソースを使用してアプリケーション接続を更新するには、管理者、デプロイヤ、デザイナー、またはオペレータのロールが割り当てられている必要があります。

アセットをエクスポートする場合、パスワード、アクセスキー、シークレットキーなどの機密フィールドはエクスポートされません。アプリケーション接続をインポートした後、アプリケーション接続と、PostmanなどのRESTクライアントで入力データとして更新するフィールドを追加できます。入力データは、接続のグローバル識別子/グローバル一意識別子（GUID）と、更新される属性のキーと値のペアで構成されます。さらに、接続のコンシューマとプロデューサに属性を提供できます。コンシューマはイベントソースを指し、プロデューサはイベントターゲットを指します。GUIDの詳細については、データ統合ヘルプの「REST API リファレンス」にある「ルックアップリソース」と「アセット検索リソース」を参照してください。

フィールド値を取得するには、次の URL を使用します。

<Informatica Intelligent Cloud Services URL>/active-bpel/asset/v1/update?assetType=AppConnection

次のサンプルに示すように、入力データを追加できます。

```
PUT <baseApiUrl>/active-bpel/asset/v1/update?assetType=AppConnection
Content-Type: application/json
Accept: application/json
INFA-SESSION-ID: <Infa-Session-ID>
{
  "connections": [
    {
      "guid": {
        "$t": "<guidvalue>"
      },
      "attributes": [
        {
          "key": {
```



フィールドが正常に更新されると、次の応答を受け取ります。

```
{
  "status": {
    "$t": "SUCCESS"
  }
}
```

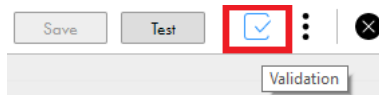
フィールドは接続の GUID に基づいて更新されます。GUID が見つからない場合は、次の応答を受け取ります。

```
{
  "status": {
    "$t": "FAILED"
  },
  "errorDetails": {
    "error": {
      "$t": "Update failed for assets"
    },
    "failedAssets": {
      "updateAssetFaultInfo": {
        "xmlns": "https://com.informatica.icrt/2021/08/updateAssets.xsd",
        "xmlns$target": "https://com.informatica.icrt/2021/08/updateAssets.xsd",
        "name": {},
        "GUID": {
          "$t": "<Incorrect guid value>"
        },
        "fault": {
          "$t": "Entry not found for GUID."
        }
      }
    }
  }
}
```

## アプリケーション接続の検証

アプリケーション接続を作成する際に [検証] パネルを使用して、検証エラーの有無を調べたりエラーの詳細を確認したりできます。

1. 次の図に示すように、ツールバーの [検証] アイコンをクリックします。



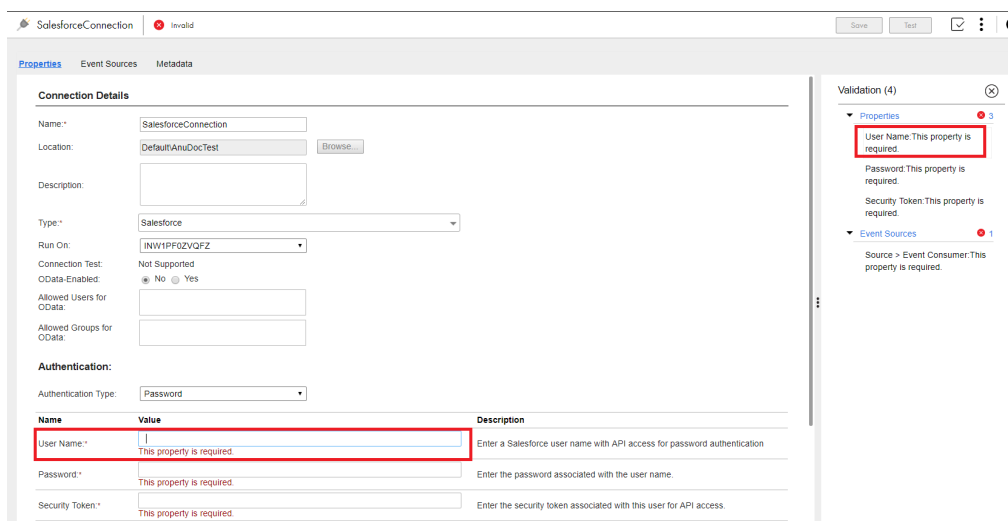
右ペインに [検証] パネルが表示されます。[検証] パネルでは、発生したタブに基づいてエラーが分類されています。

2. 次の図に示すように、タブ名に対応する矢印をクリックします。



そのタブで発生したエラーのリストが表示されます。

3. 次の図に示すように、エラー行をクリックします。



問題のあるフィールドにカーソルが移動します。問題に対処してエラーを解決します。  
エラーを修正すると、**[検証]** パネルのエラーリストが更新されます。

## アプリケーション接続の保存およびパブリッシュ

アプリケーション接続を作成したら、アプリケーション接続を保存してパブリッシュします。

複数のアプリケーション接続をまとめてパブリッシュすることもできます。詳細については、「呼び出し」を参照してください。

アプリケーション接続ページで、**[保存]** をクリックしてアプリケーション接続を保存し、**[パブリッシュ]** をクリックしてアプリケーション接続をパブリッシュします。

アプリケーション接続をパブリッシュしたら、パブリッシュ済みのメタデータを表示し、プロセスのサービスステップで接続を使用できます。

## パブリッシュ済みメタデータの表示

アプリケーション接続のパブリッシュ後、アプリケーション接続に関連付けられたオブジェクトを表示できます。

次の図で、アプリケーション接続のパブリッシュ済みメタデータを確認できます。

Salesforce ✔ Valid Save Publish Test

Properties Event Sources Event Targets Metadata

Connection last published on 2023-06-09 09:52:31 UTC and metadata last updated on 2023-06-09 09:52:31 UTC

▼ Actions (12) Find

Action Name	Description
Convert Lead	Used to convert the current Lead into an Account/Contact/Opportunity.
Create SObject	Automated step that creates an SObject
Delete SObject	Automated step that deletes an SObject
Query SObject	Automated step that returns the results of an SOQL query
Read SObject	Automated step that returns the fields of an SObject

1 - 5 of 12 Items Page 1 of 3 Items Per Page: 5

▼ Objects (1011) Find

Name	Label	Type	Description
▶ _any	Home		Represents the object which displays information t...
▶ AcceptedEventRelation	Accepted Event Relation		Salesforce entry description for: AcceptedEventR...
▶ Account	Account		Salesforce entry description for: Account
▶ AccountChangeEvent	Account Change Event		Salesforce entry description for: AccountChange...
▶ AccountCleanInfo	Account Clean Info		Salesforce entry description for: AccountCleanInfo

1 - 5 of 1011 Items Page 1 of 203 Items Per Page: 5

注:一部の接続タイプでは、パブリッシュ済みメタデータは使用できません。

## アプリケーション接続のパブリッシュ解除

パブリッシュ済みのアプリケーション接続を無効にするには、アプリケーション接続をパブリッシュ解除する必要があります。必要な変更を加えた後、そのアプリケーション接続を再びパブリッシュすると、それらの変更が反映されます。

次の図に示すように、[アクション] > [パブリッシュ解除] をクリックします。

SFAppConnection ✔ Valid Save Publish Test

Properties Event Sources Event Targets Metadata

**Connection Details**

Name:\*  (Unpublish connection to edit name)

Location:

Description:

Type:\*

Publish (Skip Metadata Refresh)  
Unpublish  
 Properties Detail...  
 Show Dependencies

## 第 8 章

# システムサービス、リスナおよびコネクタ

この章では、以下の項目について説明します。

- [OData プロバイダの使用, 255 ページ](#)
- [Salesforce システムサービスのリスナとコネクタの使用, 260 ページ](#)
- [電子メールサービスの使用, 270 ページ](#)
- [Java Message Service \(JMS\) の使用, 276 ページ](#)
- [シェルコマンド起動サービスコネクタの使用, 276 ページ](#)

## OData プロバイダの使用

OData は、REST ベースの標準化されたプロトコルであり、Web 上のデータへのアクセスを提供します。OData プロトコルを使用して、JDBC 接続で使用可能な内部データソースなど、クラウドからデータサービスにアクセスできます。

OASIS は [OData V4](#) で標準化されています。Informatica は、OData V4 を使用するバージョンを使用することをお勧めします。これがデフォルトのバージョンです。

組織は、次のようなエンドポイントで OData フィードを公開できます。

`https://[host].rt.informaticacloud.com/active-bpel/odata/[version]/[connection name]/[data source name]`

例えば、次の 2 つのエンドポイントは、*sampleparts* テーブルを公開します。

- **OData バージョン 2:** `https://[host].rt.informaticacloud.com/active-bpel/odata/v2/Parts/sampleparts`

または

- **OData バージョン 4:** `https://[host].rt.informaticacloud.com/active-bpel/odata/v4/Parts/sampleparts`

データは、XML Content-Type のサポートなどの、Atom XML フォーマットまたは JSON で利用できます。

OData スキーマが生成されるときに、次の点に注意してください。

- IID フィールドは除外されます。
- キーフィールドは、OData キー定義に追加されます。
- スキーマは、各フィールドのネイティブデータ型と同じ型を使用します。

Secure Agent で実行するように構成された接続で、OData を有効にできます。Secure Agent は、ポートを開くのではなく、Informatica Cloud サーバーへのアウトバウンド接続を開きます。すべての通信はこのサーバーを経由します。これにより、Secure Agent はオンプレミスのすべてのアプリケーションまたはデータソースにアクセスできるようになります。

**注:** OData 対応接続を Secure Agent にパブリッシュする場合、**[接続]** ページの OData URL は Informatica Cloud URL です。

Secure Agent URL は表示されません。Secure Agent URL を構築するには、Informatica Cloud URL 内の `odata/v4/Oracle` というすべてのテキストを `https://<host>:<port>/process-engine/` に置き換えます。

例えば、Informatica Cloud URL が `https://ps1w2.rt.informaticacloud.com/active-bpel/odata/v4/Oracle` の場合、Secure Agent URL は `https://localhost:7443/process-engine/odata/v4/Oracle` です。

## サポートされる OData V4 および OData V2 URI 規約

OData V4 および OData V2 の多くの URI 規約を使用して、データにアクセスできます。

### サポートされる OData バージョン 4 の規約

次の OData V4 URI 規約を使用して、データにアクセスできます。

- セクション 2。URL コンポーネント
- セクション 3 サービスルート URL
- セクション 4.1 サービスに対するモデルの対応
- セクション 4.3 エンティティの対応
  - 正規化した URL のみを使用できます
- セクション 4.4 エンティティ間の参照の対応
- セクション 4.6 プロパティの対応
- セクション 4.7 プロパティ値の対応
- セクション 5 クエリオプション
- セクション 5.1 システムクエリオプション
- セクション 5.1.1 システムクエリオプション\$filter
  - セクション 5.1.1.1 論理演算子
    - セクション 5.1.1.1.1 等しい
    - セクション 5.1.1.1.2 等しくない
    - セクション 5.1.1.1.3 より大きい
    - セクション 5.1.1.1.4 以上
    - セクション 5.1.1.1.5 より小さい
    - セクション 5.1.1.1.6 以下
    - セクション 5.1.1.1.7 AND
    - セクション 5.1.1.1.8 OR
  - セクション 5.1.1.4.1 contains
  - セクション 5.1.1.4.2 endswith
  - セクション 5.1.1.4.3 startswith
  - セクション 5.1.1.6 リテラル



- セクション 5.1.1.6.1 プリミティブリテラル
- セクション 5.1.3 システムクエリオプション\$select
- セクション 5.1.4 システムクエリオプション\$orderby
  - \$orderby は JDBC コネクタでのみ使用できます。
- セクション 5.1.5 システムクエリオプション\$stop および\$skip
  - \$skip は JDBC コネクタでのみ使用できます。
- セクション 5.1.6 システムクエリオプション\$count
- セクション 5.1.7 システムクエリオプション\$search
  - http://host/service/Products?\$search=blue のような単純な検索を実行できます
  - \$search と\$filter を同時に使用することはできません
- 5.1.8 システムクエリオプション\$format

URI 規約の完全な OData V4 仕様については、[here](#) を参照してください。

### サポートされる OData バージョン 2 の規約

Informatica では、OData V4 の使用をお勧めします。ただし、次の OData V2 URI 規約を使用して、データにアクセスできます。

- セクション 1。URI コンポーネント
- セクション 2。サービスルート URI
- セクション 3。リソースパス
- セクション 3.1 エントリの対応
  - リレーションと複雑なエンティティは使用できません
- セクション 3.3 サービス操作の対応
- セクション 4。クエリ文字列オプション
  - セクション 4.1 システムクエリオプション
  - セクション 4.3 トップシステムクエリオプション (\$top)
  - セクション 4.5 フィルタシステムクエリオプション (\$filter)
    - \$filter オプションでは、「endswith」と「startswith」のみを使用できます。
  - セクション 4.7 フォーマットシステムクエリオプション (\$format)
  - セクション 4.8 選択システムクエリオプション (\$select)
  - セクション 4.9 インラインカウントシステムクエリオプション (\$inlinecount)

URI 規約の完全な OData V2 仕様のリストについては、[here](#) を参照してください。

## カスタム複合キー

OData を使用して JDBC コネクタにアクセスすると、カスタム複合キーを定義および編集できます。データベースエンティティにプライマリキーがない場合は、カスタム複合キーを定義します。作成したこのカスタム複合キーを使用して、JDBC データベースで読み取り操作を実行できます。

カスタム複合キーを使用するには、OData V4 を使用する必要があります。

### プライマリキーとカスタム複合キー

プライマリキーは、オブジェクトフィールド（データベーステーブルまたはデータベースビュー内の 1 つのカラム）、またはオブジェクトフィールドのセット（データベース内の複数カラム）で、データベーステーブルま

たはデータベースビュー内の各レコードを一意に識別できます。プライマリキーを使用して、データベース内のオブジェクトを照会します。

エンティティにプライマリキーがない場合、エンティティに対してカスタム複合キーを定義できます。カスタム複合キーを作成するには、そのキーに対応するフィールドを1つまたは複数手動で選択します。カスタム複合キーはプライマリキーの代わりとなり、JDBC データベースからのデータの読み取りに使用できます。

例: 従業員レコードの JDBC データベースに接続すると、エンティティ *empaccount* にプライマリキーがないことがわかります。 *email*、*ID*、および *name* のフィールドを手動で選択して、*empaccount* のカスタム複合キーを定義できます。

作成したカスタム複合キーは、編集することもできます。存在しなくなったエンティティを参照するカスタム複合キーを定義した場合は、そのキーを削除できます。存在しなくなったフィールドがカスタムキーに含まれる場合は、これらのフィールドを削除できます。

## カスタム OData エンティティキーセクションの表示

1. OData を有効にして、JDBC 接続をパブリッシュします。
2. [パブリッシュ済みメタデータ] タブをクリックします。

新しいセクション [OData カスタムエンティティキー] に、次の情報が入力されています。

- エラーメッセージ (ある場合)。プライマリキーのないエンティティや、存在しなくなったエンティティに割り当てられているカスタムキーなどの情報が含まれます。
- エンティティとそれに対応するキーフィールドのリスト。

3. [カスタムキーの編集] をクリックします。

[OData カスタムエンティティキー] ダイアログボックスが表示されます。

次の図は、[OData カスタムエンティティキー] ダイアログボックスを示します。

Custom OData Entity Keys

Search:  Entity: booknopk Search:

Is Key	Name	Label	Type
<input type="checkbox"/>	Author	Author	string
<input type="checkbox"/>	Price	Price	int
<input type="checkbox"/>	Size	Size	string
<input checked="" type="checkbox"/>	Title	Title	string
<input checked="" type="checkbox"/>	id	id	int

Key fields: Title, id

Apply Cancel

左側に、次の3種類のエンティティが表示されます。

- プライマリキーのないエンティティ。
- 存在しないがカスタムキーがマッピングされたままになっているエンティティ。

- プライマリキーが定義されていて、カスタム複合キーを必要としないエンティティ。
- 右側には、マッピングしてカスタム複合キーを作成できるフィールドのリストが表示されます。

## OData カスタム複合キーの作成

OData カスタム複合キーを作成するには、プライマリキーとして使用するエンティティを選択します。

1. **[カスタムキーの編集]** をクリックします。
2. エンティティのリストから、エンティティを選択します。
3. カスタムキーとして使用するフィールドを選択します。複数のフィールドを選択できます。

**重要:** データを分析して、データベースのテーブルまたはビューで各レコードを一意に識別するフィールドを選択するようにします。例えば、フィールド **FirstName** と **LastName** を使用してカスタム複合キーを作成する場合は、姓と名が同じ従業員を 2 人含めることはできません。

4. **[適用]** をクリックします。
- 作成したカスタム複合キーが、[OData カスタムエンティティキー] のリストに表示されます。

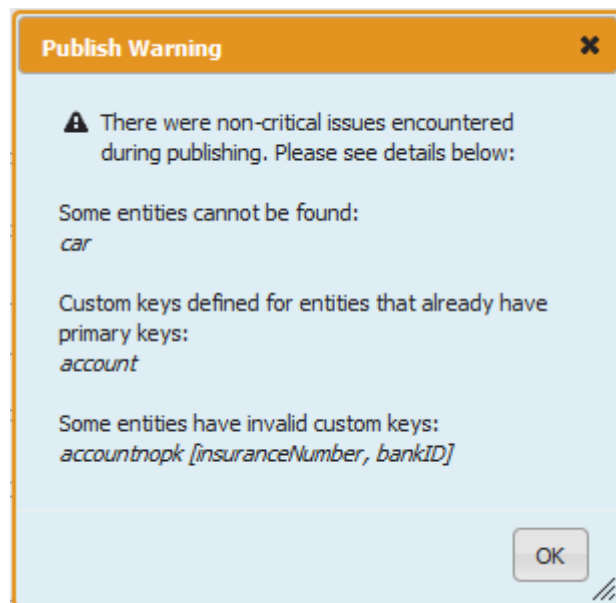
## OData カスタム複合キーの編集

OData カスタム複合キーを編集および削除できます。

## パブリッシュの検証

接続をパブリッシュするときに、Informatica Process Designer は作成したカスタム複合キーを確認し、潜在的な問題が見つかった場合は警告を表示します。

次の図に、警告メッセージの例を示します。



# Salesforce システムサービスのリスナとコネクタの使用

## Salesforce 送信メッセージ

Salesforce 送信メッセージ (OBM) を使用して、アプリケーションの統合プロセスをトリガします。

例えば、取引先の Salesforce オブジェクトへの変更が CreateNewID プロセスをトリガするように、OBM を設計します。

Salesforce OBM を作成するには、以下のタスクを実行します。

1. Informatica Cloud Real Time for Salesforce パッケージ (管理パッケージ) をダウンロードしてインストールします。詳細については、*Salesforce Managed Package* のセクションの *Salesforce Managed Package* のインストールに関するトピックを参照してください。
2. Informatica Intelligent Cloud Services<sup>SM</sup>にログインし、アプリケーションの統合サービスを開きます。
3. Salesforce OBM でトリガするプロセスを作成してパブリッシュします。
4. Salesforce 開発者アカウントにログインして、Salesforce OBM を作成します。

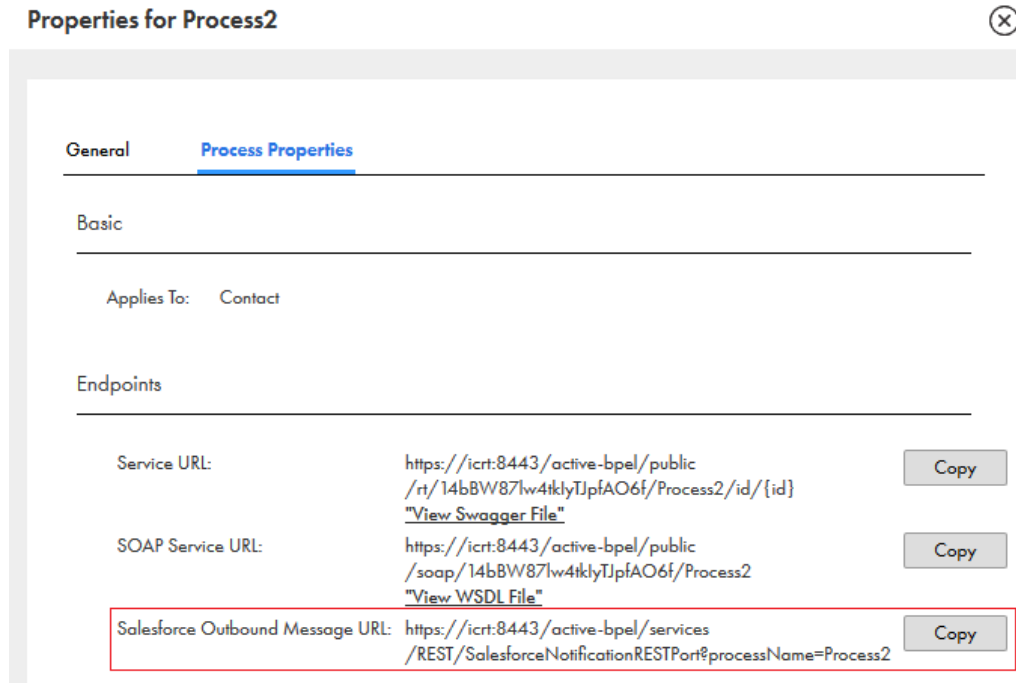
## Salesforce OBM URL

Salesforce OBM からプロセスを起動すると、Salesforce は、Salesforce OBM URL と呼ばれる特定のエンドポイントにメッセージを送信します。

プロセスの Salesforce OBM URL を表示するには、次の手順を実行します。

1. アプリケーションの統合で、Salesforce 接続を作成し、接続の名前に「Salesforce」とサフィックスを付けます。  
例えば、接続に「AccountDetails-Salesforce」または「TestSalesforce」と名前を付けて、手順 2 から 5 を実行して、Salesforce OBM URL を表示します。  
**注:** 「Salesforce」を名前の末尾以外の場所で使用した場合、Salesforce OBM は表示できません。
2. Salesforce 接続を保存してパブリッシュします。
3. Salesforce 接続を使用するプロセスを作成、保存、およびパブリッシュします。
4. プロセスの **【参照】** ページまたは **Process Designer** から、**【アクション】** > **【プロパティ】** をクリックします。

5. 表示される【プロパティ】ウィンドウで、【プロセスのプロパティ】タブをクリックして Salesforce OBM URL を表示します。  
次の図は、Salesforce OBM URL が表示された【プロセスのプロパティ】タブのサンプルを示します。

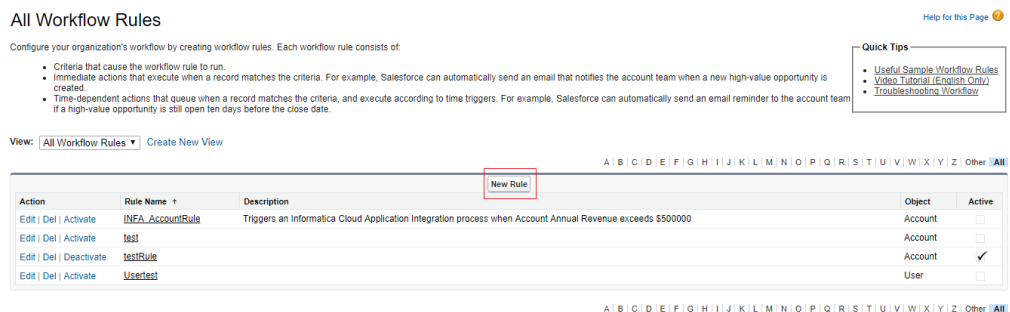


## Salesforce OBM の作成

Salesforce OBM を作成するには、Salesforce の開発者アカウントにログインし、ワークフローを作成して OBM を定義します。

Salesforce OBM を作成してアプリケーションの統合プロセスをトリガする前に、次のタスクを実行する必要があります。

- OBM を作成する Salesforce 組織に、管理パッケージをインストールします。
  - Salesforce OBM でトリガするアプリケーションの統合プロセスを作成してパブリッシュします。
1. Salesforce 開発者アカウントにログインします。
  2. 【作成】 > 【ワークフローと承認】 > 【ワークフロールール】 の順に移動します。
  3. 【新しいルール】 をクリックします。



4. **【新しいワークフロールール】** ページで、ルールを適用するオブジェクトを選択し、次に **【次へ】** をクリックします。

5. **【ワークフロールールの設定】** ページで、ルールの名前と説明を入力し、評価とルールの条件を設定して、**【保存して次へ進む】** をクリックします。

6. **【ワークフローアクションの追加】** > **【新規送信メッセージ】** の順に移動します。
7. **【送信メッセージの設定】** ページで、**【エンドポイント URL】** などの必要な情報を入力します。エンドポイント URL の取得方法の詳細については、[Salesforce OBM URL \(ページ 260\)](#) を参照してください。

8. **【保存】** をクリックします。

Salesforce OBM の作成が完了しました。作成した Salesforce OBM が、設定した条件に従ってアプリケーションの統合プロセスをトリガします。

Salesforce OBM の設定に関する詳細については、Salesforce のマニュアルを参照してください。

## 複数の Salesforce 組織から単一の IICS 組織への OBM の設定

Salesforce Managed Package は、デフォルトでシングルサインオンメカニズムを使用します。Salesforce 組織と対応する Informatica Cloud 組織との関係は 1 対 1 であることが必要です。ただし、単一の Informatica Cloud 組織で実行される特定のプロセス向けの OBM を、複数の Salesforce 組織が生成できるようにすることが必要な場合があります。

これを行うには、authType クエリパラメータを使用して Salesforce OBM を設定します。authType パラメータは、ユーザーアカウントの設定を通じて、Informatica Cloud 組織と複数の Salesforce 組織との間の関係を使用および確立するための認証のタイプを表します。

**注:** authType パラメータを使用して、複数の Salesforce 組織が同じプロセスを使用するようにすることはできません。Salesforce 組織と OBM の送信先となる特定のプロセスとの間の関係は、1 対 1 であることが必要です。

authType クエリパラメータを使用して Salesforce OBM を有効にするには、次の手順を実行します。

1. OBM を送信する Salesforce 組織ごとに、専用の Salesforce ユーザーアカウントを作成します。Informatica Cloud Real Time for Salesforce Managed Package がこれらの組織にインストールされていることを確認します。ただし、パッケージを設定して、Informatica Cloud Real Time のホスト URL を指定する必要はありません。
2. Salesforce ユーザーを特定の Informatica Cloud ユーザーに関連付けるには、次の手順を実行します。
  - a. Informatica Intelligent Cloud Services (IICS) にログインし、**[管理者]** > **[ユーザー]** を選択します。
  - b. Salesforce ユーザーに関連付ける Informatica Cloud ユーザーに移動します。
  - c. **[Salesforce ユーザー名]** および **[Salesforce ユーザー名の確認]** フィールドに、手順 1 で作成したユーザー名を入力します。
3. authType=ICSUser クエリパラメータを使用するように Salesforce の OBM を設定します。このパラメータは、Informatica Cloud で Salesforce セッション ID とセッション URL を検証するときに使用する認証のタイプを示します。クエリパラメータはリテラル値です。例えば、次のような URL を使用できます。  
`https://use4-cai.dm-us.informaticacloud.com/active-bpel/services/REST/SalesforceNotificationRESTPort?processName=TestOuboundMessage&authType=ICSUser`  
**注:** authType パラメータは、プロセスに対してのみ使用できます。
4. **[Salesforce 送信メッセージ]** 定義で、次の手順を実行します。
  - a. **[次のユーザーとして送信]** フィールドに Salesforce ユーザー名を入力します。これは、メッセージの送信中に使用されるユーザー名です。
  - b. **[セッション ID を含める]** を選択します。
  - c. 実行時に、手順 4a で指定したユーザー名が、手順 2c で入力した Salesforce ユーザー名と一致することを確認します。また、プロセスが組織内に存在することが必要です。

**注:** authType=ICSUser クエリパラメータを使用する場合、OBM を送信できるのは、1 つの Salesforce 組織につき 1 ユーザーのみです。別の Salesforce ユーザーが、すべてのユーザーロールを備えた対応するユーザーを Informatica Cloud 組織内に持つことはできますが、このユーザーから送信された OBM はプロセスをトリガしません。Salesforce の **[OBM モニタリング]** ページでは、新しく作成されたユーザーから送信された OBM について HTTP エラー (403 Forbidden) が表示されます。OBM は、同じプロセスまたは別のプロセスをトリガする可能性があります。

OBM URL に組織 ID が含まれていない場合でも、Salesforce OBM がそれぞれの Cloud アプリケーション統合組織にどのようにルーティングされるかに関する詳細については、次のコミュニティ記事を参照してください:

<https://knowledge.informatica.com/s/article/513595>。

## 単一の Salesforce 組織からさまざまな IICS 組織への OBM の設定

単一の Salesforce アカウントを使用して、さまざまな Informatica Intelligent Cloud Services (IICS) 組織にあるさまざまなプロセスを実行できます。

Salesforce アカウント S があり、この Salesforce アカウントを使用して、IICS 組織 A で作成されたプロセス A と、IICS 組織 B で作成されたプロセス B をトリガするとします。

### Salesforce アカウント S を使用して IICS 組織 A のプロセス A をトリガする

1. Salesforce アカウント S にログインし、Informatica Cloud Real Time for Salesforce パッケージをインストールします。この Managed Package は、Informatica Intelligent Cloud Services を Salesforce アカウントに接続します。
2. **【ガイド設定】** ページで、Informatica Cloud アプリケーション統合のホスト URL を次の形式で入力します。  
<Salesforce のポッド固有の Cloud Application URL>, <Informatica の組織 ID>
3. Salesforce 接続を作成します。
4. Salesforce 開発者アカウントにログインし、ワークフロールールを作成して Salesforce オブジェクトを適用します。IICS 組織 A のプロセス A を呼び出すための Salesforce OBM を作成します。
5. ワークフロールールをトリガします。  
IICS 組織 A のプロセス A が、Salesforce アカウント S を使用してトリガされます。

### Salesforce アカウント S を使用して IICS 組織 B のプロセス B をトリガする

1. IICS にログインし、**【管理者】 > 【ユーザー】 > 【ユーザーの追加】** を選択して、次の図に示すように、Salesforce 認証を使用して IICS 組織 B に新しいユーザーを作成します。

Enabled	Group Name	Description	Enabled	Role Name	Description
<input type="checkbox"/>	group		<input type="checkbox"/>	Admin	Role for performing administrat...
<input type="checkbox"/>	HTAdmin		<input type="checkbox"/>	AEgorova_Roless	A role for me.

ユーザー名と、ユーザーアカウントを有効化するための確認コードが記載された電子メールが届きます。

**注:** ユーザー名は、Salesforce アカウントのユーザー名、つまり S と同じである必要があります。

2. Salesforce 組織 S にログインします。新しい Salesforce Managed Package をインストールするには、次の URL にアクセスします。

<https://login.salesforce.com/packaging/installPackage.apexp?p0=04t3A000001AK3E>



**Install Informatica Cloud**  
By Informatica Corp

**Installation Complete!**

Done

App Name	Publisher	Version Name	Version Number
Informatica Cloud	Informatica Corp	IICS R30	5.0

インストールが完了すると、次の図に示すように、インストールされた Managed Package が [インストール済みパッケージ] の下に表示されます。

Installed Packages

On AppExchange you can browse, test drive, download, and install pre-built apps and components right into your salesforce.com environment. [Learn More about Installing Packages.](#)

Apps and components are installed in packages. Any custom apps, tabs, and custom objects are initially marked as "Development" and are not deployed to your users. This allows you to test and customize before deploying. You can deploy the components individually using the other features in setup or as a group by clicking Deploy.

Depending on the links next to an installed package, you can take different actions from this page.

To remove a package, click **Uninstall**. To manage your package licenses, click **Manage Licenses**.

Action	Package Name	Publisher	Version Number	Namespace Prefix	Install Date	Links	Apps	Tab	Objects	AppExchange Ready
Uninstall	Informatica Cloud	Informatica Corp	5.0	informatica	9/8/2022 12:02 AM		1	1	0	Passed
Uninstall	Salesforce Connected Apps	Salesforce.com	1.7	sfc_com_apps	3/29/2019 4:10 PM	✓	0	0	0	Not Passed
Uninstall	Informatica Cloud Real Time for Salesforce	Informatica	1.24	ict	2/25/2019 10:51 PM	✓	1	3	5	Not Passed

Uninstalled Packages

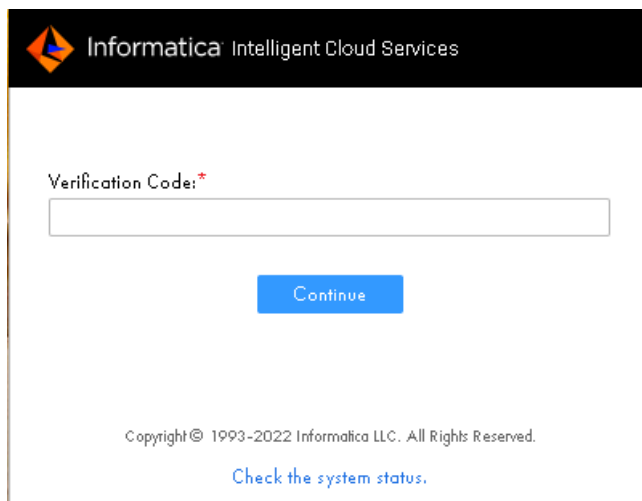
No uninstalled package data archives

3. ページ上部の [すべてのタブ] をクリックします。



[すべてのタブ] ページには、次の図に示すように [統合] タブが表示されます。

4. **【統合】** タブをクリックし、次に **【シングルサインオン】** をクリックします。  
<https://dm-us.informaticacloud.com> URL にリダイレクトされ、次の図に示すように、IICS のユーザー確認ページが表示されます。



**注:** シングルサインオンを確認するには、Salesforce ユーザーアカウントが <https://dm-us.informaticacloud.com> URL 内の任意の POD (Point of Deployment) に存在している必要があります。

5. 電子メールで受け取った確認コードを入力します。  
確認が成功すると、IICS アカウントにリダイレクトされます。
6. Salesforce アカウント S を使用してワークフロールールを作成し、Salesforce オブジェクトを適用します。IICS 組織 B のプロセス B を呼び出す OBM を作成します。authType=ICSUser クエリパラメータを使用するように Salesforce の OBM を設定します。このパラメータは、Informatica Cloud で Salesforce セッション ID とセッション URL を検証するときに使用する認証のタイプを示します。例えば、次のような URL を使用できます。

```
https://use4-cai.dm-us.informaticacloud.com/active-bpel/services/REST/SalesforceNotificationRESTPort?processName=TestOutboundMessage&authType=ICSUser
```

**注:** authType パラメータは、プロセスに対してのみ使用できます。

7. ワークフロールールをトリガします。  
IICS 組織 B のプロセス B も、Salesforce アカウント S を使用してトリガされます。

Salesforce Managed Package のインストールに関する詳細については、『*Salesforce およびアプリケーション統合*』を参照してください。

## Salesforce 接続の使用

Salesforce Managed Package を使用している場合、Salesforce コネクタを使用すると、次の認証方式のいずれかを使用して Salesforce 組織への接続を定義できます。

- OAuth 接続
- OAuth JSON Web トークン (JWT) 接続
- パスワード/セキュリティトークン認証

Salesforce 接続を新しく作成するときに、**【認証タイプ】** を選択して、関連する接続プロパティを指定します。

アプリケーション統合では、新しい Salesforce 接続を作成する際のパスワード認証および OAuth ベースの認証用に、Salesforce API バージョン 56.0 がサポートしています。バージョン 56.0 では、さまざまなデータオブジェクトとメタデータタイプが提供されています。

## OAuth 接続

Salesforce は、OAuth をサポートし、API を介して Salesforce.com にアクセスすることを許可します。OAuth は、セキュアな API 承認に使用できる標準プロトコルです。OAuth を使用すると、ユーザーは Salesforce の資格情報を公開する必要がなく、Salesforce 管理者はいつでもコンシューマのアクセスを取り消すことができます。

アプリケーション統合では、新しい Salesforce 接続を作成する際のパスワード認証および OAuth ベースの認証用に、Salesforce API バージョン 56.0 がサポートしています。バージョン 56.0 では、さまざまなデータオブジェクトとメタデータタイプが提供されています。例えば、個々のアカウントに対応する連絡先の請求先住所または配送先住所を表す ContactPointAddress オブジェクトを使用できます。

**注:** Salesforce API バージョン 56.0 によって提供されているさまざまなデータオブジェクトおよびメタデータタイプを Salesforce アプリケーション接続に使用するには、Salesforce にインストールされている管理ページをアップグレードする必要があります。Salesforce オブジェクトとサービスの詳細については、*Salesforce コネクタガイド*を参照してください。

Salesforce 接続の認証タイプとして OAuth を有効にするには、次の手順を実行します。

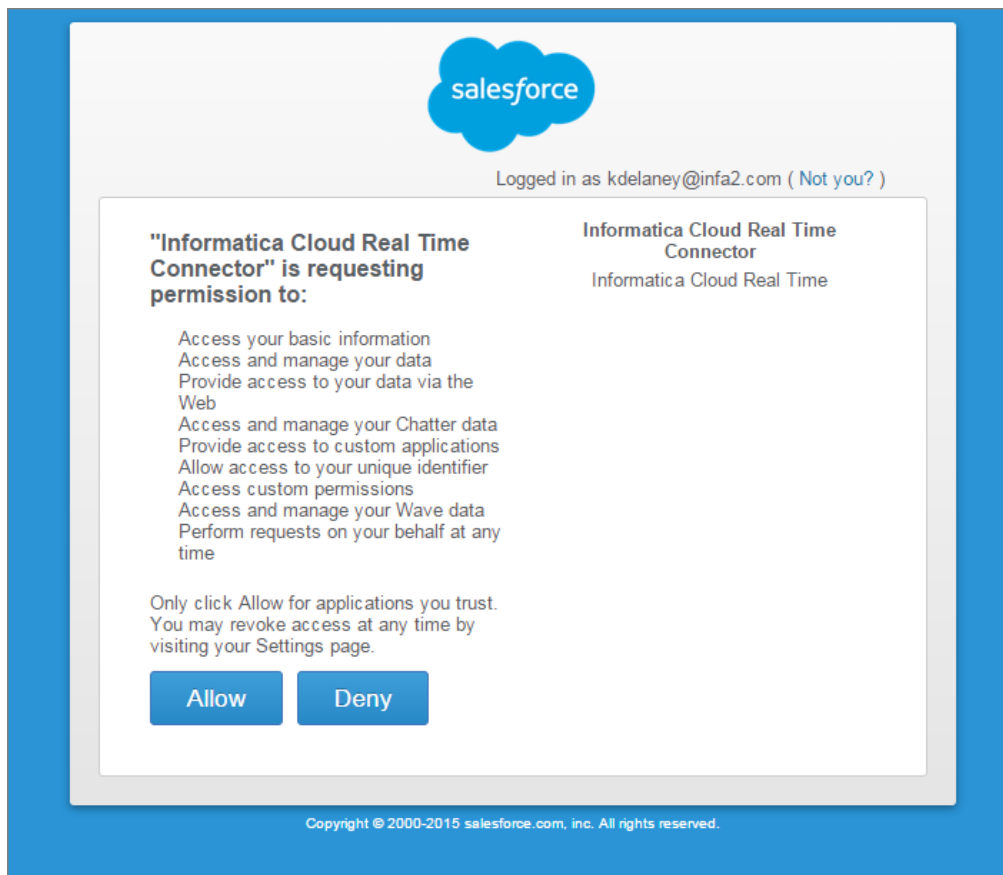
1. アプリケーション統合で、**[新規]** をクリックします。
2. **[新しいアセット]** ダイアログボックスで、**[アプリケーション接続]** > **[作成]** をクリックします。**[アプリケーション接続]** ページが表示されます。
3. **[認証]** セクションで、リストから **[OAuth]** を選択します。

Authentication:		
Name	Value	Description
Authentication Type:	<input type="text" value="OAuth"/>	
Authorization URL*	<input type="text" value="https://login.salesforce.com/services/oauth2/authorize"/>	Enter the Salesforce OAuth authorization URL. Default value for production is https://login.salesforce.com/services/oauth2/authorize. For Sandbox, use https://test.salesforce.com/services/oauth2/authorize.
Token Request URL*	<input type="text" value="https://login.salesforce.com/services/oauth2/token"/>	Enter the OAuth token request URL. For production, use https://login.salesforce.com/services/oauth2/token. For sandbox, use https://test.salesforce.com/services/oauth2/token.
Session Duration:	<input type="text" value="60"/>	Enter the number of minutes to wait before refreshing the session. Default is 60 minutes.
Authorization Status:	<input type="text" value="Not yet authorized"/>	Indicates the current status and the last time that authorization was completed.
Authorize Access:	<input type="button" value="Authorize"/>	Click to initiate the authorization workflow using OAuth.

4. 次の必須フィールドの画面で指示されるように、製品またはテスト URL を入力します。
  - **[承認 URL]**。Salesforce は、認証を処理する専用の URL を提供します。この接続でアクセスするための、本番環境またはテスト環境の URL を入力します。
  - **[トークン要求 URL]**。Salesforce は、トークン要求を処理する専用の URL を提供します。この接続でアクセスするための、本番環境またはテスト環境の URL を入力します。
5. **[セッション期間]** ボックスに、この接続用に開始した各セッションを維持する時間（分）を入力します。デフォルトは 60 分です。
6. **[承認]** をクリックして、承認プロセスを開始します。

**注:** 承認プロセスを正常に起動するには、ブラウザのポップアップを有効にする必要があります。開始した承認プロセスは、ダイアログボックスを閉じると、完了する前にキャンセルすることができます。その場合、前の状態（承認済みまたは非承認）は変更されません。

7. 入力が必要であれば、Salesforce のログインページでユーザー名とパスワードを入力し、**[Salesforce にログイン]** をクリックします。ログインが検証されると、次のページが表示されます。



8. **[許可]** をクリックして、承認を完了します。Salesforce 接続プロパティでは、**[承認ステータス]** メッセージが表示されます。メッセージには、接続を承認した最後のユーザーの名前と承認の時刻が含まれます。

**注:** 承認プロセスは3分以内に完了する必要があります。そうならなかった場合、プロセスはタイムアウトになります。その場合は、接続プロパティに戻って、再び **[承認]** をクリックします。

OAuth アクセスが Salesforce で取り消されていない限り、接続がパブリッシュされている限り、承認はアクティブなままになります。セッションの有効期限が切れると、バックグラウンドで自動的に更新されます。

## Salesforce Connected App の設定

Salesforce 管理者が有効にした OAuth アクセスのタイプに応じて、自己承認または事前承認することができる場合があります。このオプションは、次の図に示す Salesforce 接続アプリケーション設定によって決まります。

The screenshot displays the 'Connected App Edit' interface for 'Informatica Cloud Real Time for Salesforce Connected App'. The page is divided into several sections:

- Basic Information:** Includes fields for 'Start URL' and 'Mobile Start URL', both marked as required information.
- OAuth policies:** Features dropdown menus for 'Permitted Users' (with options like 'All users may self-authorize' and 'Admin approved users are pre-authorized') and 'IP Relaxation' (with an option 'Enforce IP restrictions'). A 'Refresh Token Policy' is also set to 'Refresh token is valid until revoked'.
- Session Policies:** Includes a 'Timeout Value' dropdown set to 'None' and a checkbox for 'High assurance session required'.
- User Provisioning Settings:** Contains a checkbox for 'Enable User Provisioning'.

At the bottom, there are 'Save' and 'Cancel' buttons.

詳細については、Salesforce のマニュアルを参照してください。

## OAuth JWT 接続

Salesforce 接続で OAuth JSON Web トークン (JWT) 認証を設定して、Salesforce に接続できます。

OAuth JWT 認証を使用して、情報を交換するたびにログインなしでデータにアクセスすることをサーバーに許可します。OAuth JWT 認証では、証明書を使用して JWT 要求の署名が行われ、明示的なユーザー操作は必要ありません。

OAuth JWT 認証タイプを選択する場合は、次のプロパティを設定します。

- **ユーザー名。** 接続アプリケーションへのアクセスが可能な Salesforce ユーザー名。
- **キーストアファイル。** PKCS12 形式のキーストアファイル。
- **キーストアのパスワード**キーストアファイルにアクセスするためのパスワード。
- **セッション期間。** セッションが期限切れになるまでの分数。
- **コンシューマキー。** Salesforce 接続アプリケーションに関連付けられたコンシューマキー。
- **トークン要求 URL。** OAuth トークン要求の URL。
- **オーディエンス。** トークンの対象ユーザーの認証サーバー URL。

Salesforce 接続での OAuth JWT 認証の詳細については、『Salesforce コネクタガイド』を参照してください。

## パスワード/セキュリティトークン認証

パスワード認証タイプを選択するときに、これらのプロパティを指定します。

- **ユーザー名:** 接続に使用される API アクセス用のユーザー名。
- **パスワード:** この接続に関連付けられているユーザーのパスワード。
- **セキュリティトークン:** このユーザーに API アクセスを提供するセキュリティトークン。
- **サービス URL:** ログインに使用する Salesforce URL。Soap/c WSDL (Soap/u ではない) を指定することに注意してください。

実行時に、提供するパスワードとセキュリティトークンに基づいて、各接続が有効になります。

## 電子メールサービスの使用

### 電子メール接続プロパティ

電子メール接続を定義して、電子メールサーバーから指定された受信者のリストに電子メールを送信できます。電子メールアプリケーション接続には、OAuth 2.0 認証またはパスワードベースの認証を使用できます。OAuth 2.0 認証は、Microsoft Outlook 電子メールアカウントにのみ使用できます。

場合によっては、複数の電子メール接続を定義する必要があります。例えば、大学内の各学部に異なる SMTP ホストを使用することがあります。または、各ドメインが固有の SMTP 設定を持つ必要がある複数のドメインを設けることがあります。各 SMTP ホストに、個別の接続が必要です。

**[IICS Cloud Application Integration 電子メール (使用ライセンスあり)]** を接続タイプとして選択した場合、接続作成ページで電子メール固有の接続プロパティを設定できます。

パスワードベースの認証を使用して電子メール接続を作成するには、接続作成ページで次のプロパティを設定します。

プロパティ	説明
認証	必須。電子メール接続でパスワードベースの認証を使用する必要があるかどうかを指定します。 [有効化] を選択した場合、電子メールコネクタは、電子メール接続プロパティに入力したユーザー名とパスワードを認証します。 [無効化] を選択した場合、電子メールコネクタは認証を実行せず、電子メール接続プロパティに入力したユーザー名とパスワードを無視します。 デフォルトは [有効化] です。
ユーザー名	パスワードベースの認証を有効にする場合は必須です。パスワードベースの認証を無効にすると、電子メールコネクタは認証を実行せず、このフィールドに入力した値を無視します。 電子メールサーバーにログインするためのユーザー名を入力します。ユーザー名は、アカウント名または電子メールアドレスのいずれかです。例: notifyme@mydomain.com 注: 環境内の SMTP をパスワードベースの認証なしで動作するように設定した場合、ユーザー名とパスワードを入力しないでください。
パスワード	パスワードベースの認証を有効にする場合は必須です。パスワードベースの認証を無効にすると、電子メールコネクタは認証を実行せず、このフィールドに入力した値を無視します。 電子メールアドレスのパスワードを入力します。 注: 環境内の SMTP をパスワードベースの認証なしで動作するように設定した場合、ユーザー名とパスワードを入力しないでください。

プロパティ	説明
セキュリティ	<p>必須。電子メール接続で使用する必要のあるセキュリティプロトコルを指定します。</p> <p>次のいずれかの値を選択します。</p> <ul style="list-style-type: none"> <li>- なし。電子メール接続はセキュリティプロトコルを使用しません。</li> <li>- TLS。電子メール接続は TLS プロトコルを使用します。</li> <li>- SSL。電子メール接続は SSL プロトコルを使用します。</li> </ul> <p>デフォルトは <b>【なし】</b> です。</p>
接続タイムアウト	<p>オプション。電子メールサーバーへの接続を試行するときに待機するミリ秒数を指定します。</p> <p>デフォルトは 30000 ミリ秒です。</p> <p>電子メール接続の使用中にタイムアウトが発生した場合、ネットワークおよびファイアウォールの設定を確認します。</p>

OAuth 認証を使用して電子メール接続を作成するには、接続作成ページで次のプロパティを設定します。

プロパティ	説明
認証 URL	<p>必須。ユーザー要求の認証に使用する電子メールサービスの OAuth 認証 URL を入力します。</p> <p>例: <a href="https://login.microsoftonline.com/xxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxx/oauth2/v2.0/authorize">https://login.microsoftonline.com/xxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxx/oauth2/v2.0/authorize</a></p>
トークン要求 URL	<p>必須。トークン要求を処理する OAuth トークン要求 URL を入力します。</p> <p>例: <a href="https://login.microsoftonline.com/xxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxx/oauth2/v2.0/token">https://login.microsoftonline.com/xxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxx/oauth2/v2.0/token</a></p> <p>リフレッシュトークンの有効期限は 90 日です。ユーザーは、トークンの有効期限が切れる前に、再度認証して接続をパブリッシュする必要があります。</p>
クライアント ID	<p>必須。OAuth プロバイダからの識別子の値を指定します。</p>
クライアントシークレット	<p>必須。電子メールアプリケーションに接続するためのクライアントシークレットを入力します。</p>
スコープ	<p>必須。スコープを指定します。OAuth 認証のスコープは、アプリケーションからユーザーのアカウントへのアクセスを制限するものです。1 つのクライアントに対して複数のスコープを選択できます。複数のスコープを入力するには、各値をスペースで区切ります。</p> <p>Microsoft Outlook 電子メールアカウントの場合は、次のスコープを入力します。</p> <p><a href="https://outlook.office.com/SMTP.Send offline_access">https://outlook.office.com/SMTP.Send offline_access</a></p>
承認ステータス	<p>現在のステータス、承認したユーザーの名前、最後に承認が完了した時刻を示します。</p>
アクセスの承認	<p>OAuth を使用した承認ワークフローを開始するにはクリックします。</p>

**注:** OAuth プロバイダで、接続を作成する CAI POD のホストにリダイレクト URI を設定します。リダイレクト URI には、/oauthcallback 文字列が含まれている必要があります。例: <https://<Informatica Cloud Application Integration URL>/oauthcallback>

接続作成ページで次の共通プロパティを設定します。

プロパティ	説明
ホスト	必須。mail.mydomain.com などの電子メールサーバーの DNS 名、または、192.168.1.1 などの IP アドレスを入力します。
ポート	必須。プロセスサーバーと電子メールサーバーとの間の通信用のポートを入力します。デフォルトは 25 です。
電子メールアドレスのテスト	オプション。電子メール接続をテストするときにテストメッセージを送信する電子メールアドレスを入力します。 複数の電子メールアドレスはカンマで区切ります。
送信者電子メールアドレス	オプション。接続から送信する電子メールの【送信者】フィールドに表示する電子メールアドレスを入力します。例: no-reply@example.com

## 電子メール接続のパブリッシュ

電子メール接続を作成した後に、接続を検証、保存、およびパブリッシュできます。詳細については、「デザイン」を参照してください。

電子メール接続をパブリッシュすると、アプリケーション統合は【メタデータ】タブで次の項目を作成します。

### sendEmailService

sendEmailService は、電子メールを送信するためにプロセスまたはガイドのサービスステップで選択する必要があるアクションです。

次の図は、sendEmailService アクションを示しています。



### AttachmentInformation

AttachmentInformation オブジェクトは、電子メールで送信される添付ファイルを定義します。XQuery 添付ファイル関数を使用して添付ファイルを作成できます。添付ファイルの作成と使用の詳細については、次のリンクを参照してください: [Attachments](#)。

次の図は、AttachmentInformation オブジェクトを示しています。



MyEmailConnection | Valid

Save Test

Properties Metadata

▼ Objects

Search:

Name	Label	Type	Description
AttachmentInformation	Attachment Details		The Attachment Details object.
name	Attachment Name	string	
content	Content	attachment	
mimeType	Mime-Type	string	

## EmailTaskInput

EmailTaskInput オブジェクトは、電子メールの入力を定義します。このオブジェクトは、受信者、電子メールの件名、電子メールのコンテンツタイプ、電子メールの本文、および添付ファイル情報を定義します。

次の図は、EmailTaskInput オブジェクトを示しています。

MyEmailConnection | Valid

Save Test

Properties Metadata

▼ EmailTaskInput

Name	Label	Type	Description
EmailTaskInput	Email Input		The email Input Details object.
to	Recipient Addresses	string	
cc	CC Addresses	string	
bcc	BCC Addresses	string	
subject	Subject	string	
contentType	Content-Type	string	
body	Body	textarea	
attachments	Attachments	MyEmailConnection:AttachmentInformation[]	

## EmailTaskOutput

EmailTaskOutput オブジェクトは、sendEmailService アクションの応答を定義します。このオブジェクトは、電子メールが正常に送信されたかどうかを定義し、次の詳細も一覧表示します。

- 電子メールの送信先の電子メールアドレス。
- 有効であっても、電子メールが送信されなかった電子メールアドレス。
- 無効な電子メールアドレス。

次の図は、EmailTaskOutput オブジェクトを示しています。

MyEmailConnection | Valid

Save Test

Properties Metadata

Search:

Name	Label	Type	Description
▶ AttachmentInformation	Attachment Details		The Attachment Details object.
▶ EmailTaskInput	Email Input		The email Input Details object.
▼ EmailTaskOutput	Email Output		The email Output Details object.
validSentAddress	Valid Sent Email Address	string	
validUnsentAddress	Valid Unsent Email Address	string	
invalidAddress	Invalid Email Address	string	
status	Status	string	

## プロセスまたはガイドでの電子メールオブジェクトの使用

プロセスまたはガイドで電子メールオブジェクトを使用するには、次の手順を実行します。

1. 電子メールオブジェクトを割り当てるフィールドを作成します。
2. **[タイプ]** リストで **[タイプをさらに表示]** を選択します。
3. **[カテゴリ]** リストで **[定義された接続タイプ]** を選択します。
4. **[接続]** リストで、電子メール接続を選択します。

5. 次の図に示すように、【タイプ】セクションで、必要な電子メールオブジェクトを選択します。

Category: \* Connection defined Types

Connection: EmailConnectionAndProcessForCAI > MyEmailConnection

The Field can be one of several types.

Types

Attachment Details

Email Input

Email Output

Allow a list of objects of this type.

OK Cancel

次に、プロセスまたはガイドの電子メールオブジェクトフィールドに値を割り当てることができます。

## プロセスまたはガイドでの電子メール接続の使用

電子メール接続を作成した後、プロセスまたはガイドのサービスステップで使用できます。

プロセスまたはガイドで電子メール接続を使用するには、次の手順を実行します。

1. プロセスまたはガイドにサービスステップを追加します。
2. 【サービス】 タブをクリックします。
3. 【サービスタイプ】 リストから、【接続】 を選択します。
4. 【接続】 リストから、作成した電子メール接続を参照して選択します。
5. 【アクション】 リストで、電子メール接続がパブリッシュされたときに作成された **sendEmailService** アクションを選択します。アプリケーション統合は、入力フィールドの下の **emailTaskInput** オブジェクトと出力フィールドの下の **emailTaskOutput** オブジェクトを読み込みます。次の図は、電子メールを送信するサンプルのサービスステップを示しています。

sendEmailService Properties

General

Service Type: Connection

Connection: EmailConnectionAndProcessForCAI > MyEmailConnection

Action: sendEmailService

Description

No description

Input Fields

Name	Required	Type
emailTaskInput	<input checked="" type="checkbox"/>	Object_ID_(MyEmailConnection:EmailTaskInput)

Output Fields

Name	Type
emailTaskOutput	Object ID (MyEmailConnection:EmailTaskOutput)

6. **【入力フィールド】** タブをクリックします。
7. 電子メールメッセージのコンテンツを **【emailTaskInput】** フィールドに設定します。  
複数の受信者に電子メールを送信するには、電子メールアドレスをカンマで区切ります。
8. 必要に応じて、電子メールの添付ファイルと出力フィールドを設定します。
9. プロセスまたはガイドを検証、保存、パブリッシュ、および実行します。

## Java Message Service (JMS) の使用

Java Message Service (JMS) を使用すると、ゆるく結び付いたプロセス間で、メッセージを交換することが可能になります。メッセージングの目的での「ゆるく結び付く」とは、メッセージが互いに独立していることを意味します。プロセス間の相互作用は、同期または非同期にすることができます。

メッセージコンシューマが実行されていない間にプロセスがメッセージを送信すると、メッセージはメッセージコンシューマが使用可能になるまで JMS プロバイダのキューで待機します。これらのメッセージのコンテンツはさまざまですが、メッセージ内のデータは通常 XML で表されます。

JMS で発生するデータフローを利用すると、オンプレミスで実行されるデータベース管理システムなどのサービスを呼び出す必要がなくなります。

## シェルコマンド起動サービスコネクタの使用

シェルコマンドスクリプトを実行するための POJO サービスを呼び出す起動アクティビティを作成できます。起動アクティビティは、プロセスサーバーが提供するシェルコマンドシステムサービスに基づきます。例えば、データベースを更新するスクリプトや、ID サービスに新規ユーザーを追加するスクリプトを呼び出すことができます。シェルコマンドシステムサービスを使用するには、起動アクティビティへの入力メッセージ内で指定された作業ディレクトリにスクリプトを作成して配置する必要があります。

次のメッセージは、起動アクティビティに対する入力の例を示します。

```
<shl:execCommand xmlns:inv =
  "urn:com:activee:rt:shellcmd:services:invoke"
  xmlns:shl="urn:com:activee:rt:shellcmd:services:ishellcmdinvoker">
```

```

<inv:shellRequest>
  <inv:command>string</inv:command>
  <inv:workingDirPath>string</inv:workingDirPath>
  <inv:charset>utf-8</inv:charset>
  <inv:mergeErrorAndOutput>true</inv:mergeErrorAndOutput>
  <inv:stdin>string</inv:stdin>
</inv:shellRequest>
</shl:execCommand>

```

次の表で、メッセージの要素を説明します。

execCommand 入力要素	説明
command	(必須) 実行するスクリプトの名前。例: cmd.exe /C myScript concat('shell.bat', \$requestVar)
workingPathDir	(必須) スクリプトの実行用の作業ディレクトリ。
charset	スクリプトの文字エンコード。デフォルトは UTF-8 です。
mergeErrorandOutput	デフォルト設定は true です。true の場合、エラーと実行の出力が同じコンソールにリダイレクトされます。
stdin	スクリプトのデータ

## 第 9 章

# ヒューマンタスクの設計

意思決定が必要なビジネスプロセスでは人間の介入が必要になります。例えば、承認や例外の管理には人間のアクションが必要です。このような場合は、ヒューマンタスクを使用できます。ヒューマンタスクを完了するには人間の介入が必要です。ヒューマンタスクアセットを作成して、ビジネスプロセスに含めるユーザータスクを定義します。ヒューマンタスクの定義を作成および編集できます。

まず、アプリケーション統合でヒューマンタスクアセットを設定します。このアセットは、ヒューマンタスク、ヒューマンタスクにアクセスできるユーザーとロール、およびタスクの結果を定義します。ヒューマンタスクステップを使用して、ヒューマンタスクアセットをアプリケーション統合プロセスに追加できます。

プロセスが実行されると、生成されたヒューマンタスクがヒューマンタスクサービスのヒューマンタスクインボックスに送信されます。自分に割り当てられたロールに基づいて、ヒューマンタスクインボックスを使用して自分に割り当てられたタスクを表示し、適切なアクションを実行できます。ヒューマンタスクインボックスの詳細については、ヒューマンタスクのヘルプを参照してください。

タスクが完了すると、プロセスはコールバックを受け取り、後続のステップに進みます。

プロセスが完了すると、アプリケーション統合コンソールでヒューマンタスクステップの実行の詳細を確認できます。

### 例

ローンの承認プロセスを作成したいと考えています。このプロセスでは、要求者がローン申請を提出できるようにする必要があります。要求は、さまざまな評価基準に基づいて要求を承認または却下する資格のある銀行員またはローン代理店に割り当てられます。次に、人間のユーザーが申請内容を確認し、承認または却下します。ローンが承認されると、ローンが要求者に認可されます。

このユースケースを実現するには、Informatica Intelligent Cloud Services で次のタスクを実行します。

1. アプリケーション統合でローン承認プロセスを作成します。
2. アプリケーション統合でヒューマンタスクアセットを作成します。アセットはヒューマンタスクの詳細を定義するものです。
3. ヒューマンタスクステップをプロセスに追加して、ヒューマンタスクアセットとタスクを含めます。
4. プロセスが実行されると、タスクがヒューマンタスクサービスのヒューマンタスクインボックスに送信されます。
5. ヒューマンタスクアセットで定義されている潜在的な所有者、管理者、および関係者は、ヒューマンタスクサービスでヒューマンタスクインボックスを開くと、ヒューマンタスクを確認できます。
6. リストに掲載されている潜在的な所有者はタスクを引き受けることができます。タスクを引き受けたユーザーはタスク所有者になり、タスクに対してさらなるアクションを実行できます。
7. タスク所有者はローン申請を分析し、承認または却下して、ヒューマンタスクを完了します。
8. その後、プロセスはアプリケーション統合の次のステップに進み、ローンが要求者に認可されます。

# ヒューマンタスクのユーザーロール

ヒューマンタスクを完了するには人間の介入が必要です。アプリケーション統合でヒューマンタスクアセットを作成する場合、タスクを監視して対応するユーザーまたはロールを割り当てます。タスクに対して実行できるアクションは、特定のユーザーまたはロールに割り当てられた権限によって異なります。

ヒューマンタスクに対して次のロールを設定できます。

## タスク所有者

タスク所有者は、タスクが割り当てられて、そのタスクにアクションを実行できるユーザーです。タスク所有者は、タスクを開始、停止、一時停止、再開、リリース、および完了できます。タスク所有者はタスクにファイルを添付したり、コメントを追加したりすることもできます。タスクには必ずタスク所有者が1人います。

## 潜在的な所有者

潜在的な所有者はタスクの実際の所有者になり、タスクに対応できるユーザーグループに属することができます。タスクが生成されると、潜在的な所有者全員がタスクを見ることができます。潜在的な所有者はタスクを引き受けることができます。タスクが潜在的な所有者によって引き受けられると、その潜在的な所有者がタスク所有者になり、他の潜在的な所有者はタスクを引き受けることができなくなります。

潜在的な所有者は、タスクの引き受け、開始、停止、一時停止、再開、リリース、および完了ができます。潜在的な所有者はタスクにファイルを添付したり、コメントを追加したりすることもできます。

## 除外する所有者

除外する所有者は、ヒューマンタスクの処理から除外するユーザーまたはロールです。除外する所有者は、タスクへのアクセス、タスクの所有はできず、タスクに対してアクションを実行できません。

## 管理者

管理者は、タスクの割り当ておよび再割り当てを許可されたユーザーまたはロールです。管理者はタスクにファイルを添付したり、コメントを追加したりすることもできます。

## 関係者

関係者は、タスクにおける監督を担い、問題が発生したときに必要なアクションを実行できるユーザーまたはロールです。関係者は、タスク所有者がタスクに対して実行したアクションを監視し、タスクを引き受け、必要に応じて別のユーザーにタスクを再割り当てできます。関係者はタスクにファイルを添付したり、コメントを追加したりすることもできます。タスクには複数の関係者を含めることができます。

# ガイドレンダリングタスク

ガイドレンダリングタスクでは、タスク所有者がデータを入力してタスクを完了するようにガイドするウィザードがユーザーに表示されます。

ガイドはユーザーにデータのレビュー、入力、または確認を求める画面のセットです。アプリケーション統合でガイドを作成します。アプリケーション統合でヒューマンタスクアセットを作成する場合、タスク所有者が簡単に操作できるように、ガイドを使用してウィザードを表示することができます。

例えば、あるステップでローン申請を表示したり、顧客が要求したローン金額の承認をユーザーに求めたりする場合があります。ガイドの手順では、データを抽出して保存することによってアプリケーションとやりとりします。ガイドは、モバイルアプリまたはPCやMacなどの従来のプラットフォームで動作します。

アプリケーション統合でヒューマンタスクアセットの作成時にガイドを選択すると、インボックス内のタスクは、ガイドで設定された画面を使用してデータを表示します。入力フィールド、出力フィールド、選択したガイドの結果は、ヒューマンタスクアセットで指定する値に一致する必要があります。ヒューマンタスクの作成時にガイドを選択しない場合、タスクはデフォルトのレンダリングを使用します。

# ヒューマンタスクの作成

ヒューマンタスクアセットを使用して、ヒューマンタスクのプロパティを定義します。アセットを作成するときに、タスクの詳細、ユーザーまたはロールの割り当て、入力フィールドと出力フィールド、タスクの結果、インボックスの表示設定などのすべてのヒューマンタスク設定を指定できます。

ヒューマンタスクアセットを作成するには、次の手順を実行します。

1. 基本的なヒューマンタスクアセットの詳細を定義します。
2. タスクを引き受けて対応できる、潜在的な所有者を定義します。
3. タスクにアクセスできず対応もできない、除外する所有者を定義します。
4. タスクの割り当ておよび再割り当てができるタスク管理者を定義します。
5. タスクの監視、割り当て、再割り当てができる関係者を定義します。
6. タスクの入力フィールドと出力フィールドを設定します。
7. タスクの結果を設定します。
8. タスクのインボックスの表示設定を指定します。



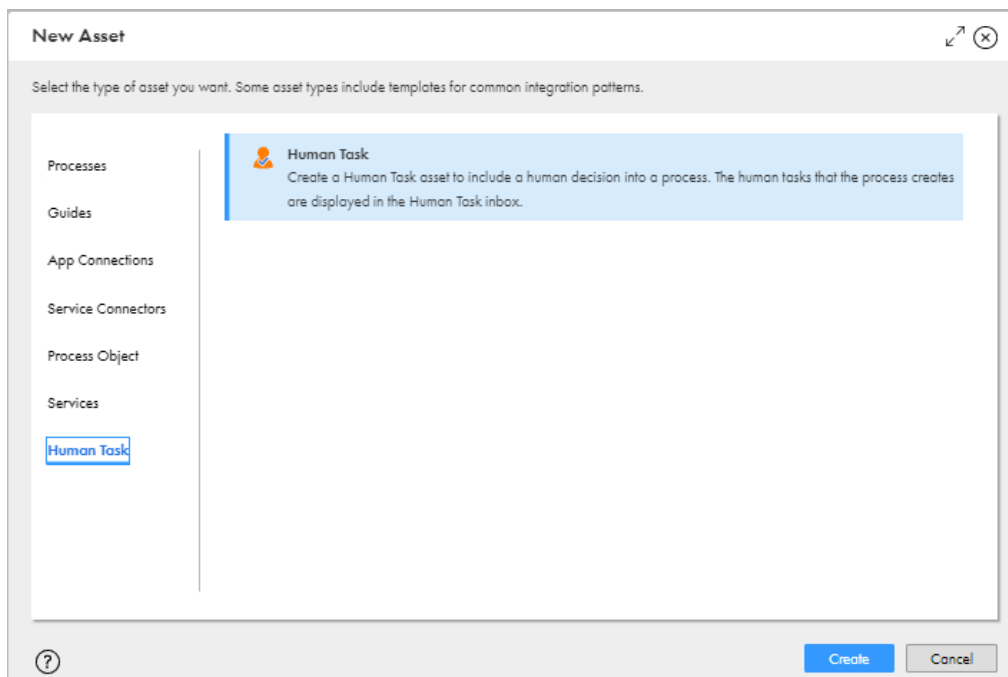
## 手順 1. ヒューマンタスクアセットを定義する

ヒューマンタスクアセットの【定義】タブでは、タスクを識別する名前、タスクのプロパティを保存する場所、タスクの説明、およびタスクの優先度を指定できます。優先度が低いタスクをユーザーがスキップできるようにタスクを設定することもできます。

ヒューマンタスクアセットの【定義】タブで指定するタスク名と説明は、ヒューマンタスクインボックスのユーザーのタスクパネルに表示されるため、ユーザーは詳細を表示しなくてもタスクをすぐに特定できます。

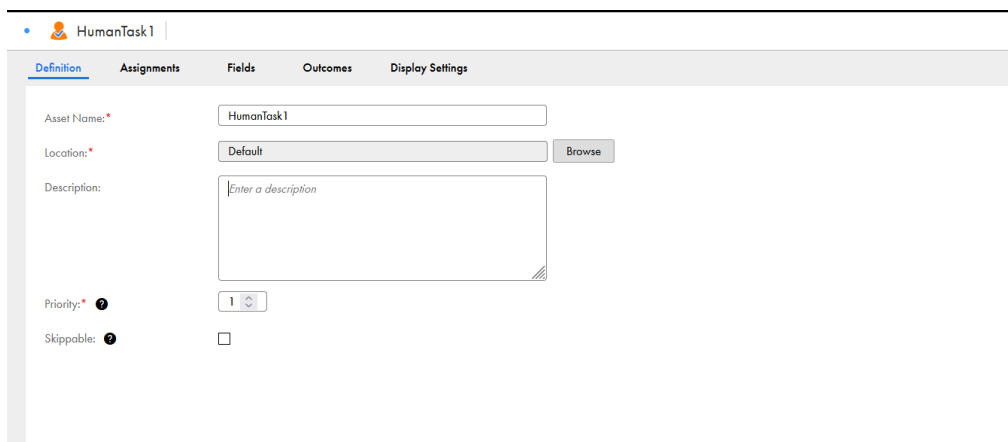
1. **アプリケーション統合**で、**新規** をクリックします。

**【新しいアセット】** ダイアログボックスが表示されます。次の図は、**【新しいアセット】** ダイアログボックスを示しています。



2. **【ヒューマンタスク】** を選択して **【作成】** をクリックします。

次の図のように、**【定義】** タブが表示されます。



3. ヒューマンタスクアセットの名前を入力します。
4. デフォルトのプロジェクトまたはフォルダの場所を使用するか、任意の場所を選択します。任意の場所を選択するには、**【参照】** をクリックし、新しい場所を選択して、**【選択】** をクリックします。
5. 必要に応じて、タスクの説明を入力できます。
6. プロセスが作成したヒューマンタスクの優先度を定義してください。小さい数字ほど、優先度が高くなります。
7. 確認者がヒューマンワークフローインボックスのタスクをスキップできるように、重要度の低いタスクには **【スキップ可能】** オプションを選択します。タスクがスキップされると、プロセスは次のステップに進みます。

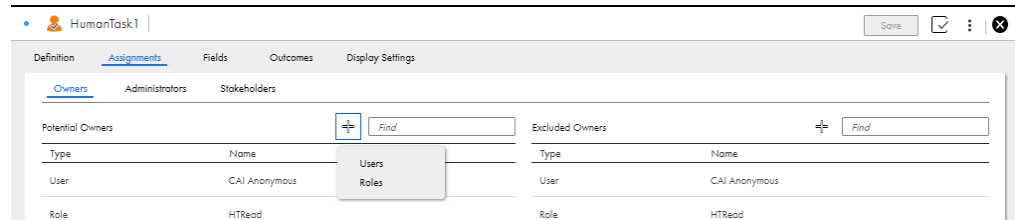
## 手順 2.潜在的な所有者を定義する

ヒューマンタスクアセットの **【割り当て】** タブで、タスクにアクセスしてアクションを実行できるすべてのユーザーとロールのリストを追加できます。潜在的な所有者のリストを **【割り当て】** > **【所有者】** タブで追加できます。潜在的な所有者はタスクの実際所有者になることができ、タスクを引き受けたあとにそのタスクに取り組むことができます。

タスクを引き受けて対応できるユーザーまたはロールを追加します。潜在的な所有者を追加しない場合、すべてのユーザーがタスクを引き受けられます。潜在的な所有者としてユーザー 1 人だけを追加する場合、設定された所有者だけにタスクが割り当てられます。

1. **【割り当て】** タブをクリックします。

次の図は、**【所有者】** タブを示しています。



2. **【所有者】** タブで、**【潜在的な所有者】** セクションの **【+】** をクリックし、リストからユーザーまたはロールを追加します。
3. 次のいずれかの手順を実行します。
  - ユーザーを追加するには、**【ユーザー】** をクリックして、リストからユーザーを選択します。**【OK】** をクリックします。
  - ロールを追加するには、**【ロール】** をクリックして、リストからロールを選択します。**【OK】** をクリックします。

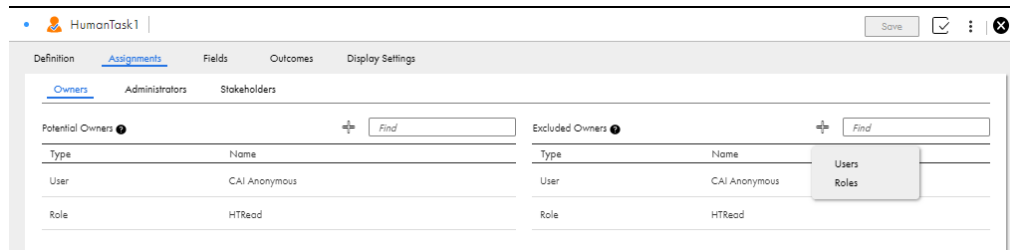
## 手順 3.除外する所有者を定義する

ヒューマンタスクアセットの【割り当て】 > 【所有者】 タブで、タスクへのアクセス、タスクの所有、またはタスクに対するアクションの実行ができない、除外する所有者のリストを追加できます。

タスク引き受けから除外したい所有者を追加します。潜在的な所有者および除外する所有者の両方としてユーザーまたはロールを追加する場合、そのユーザーまたはロールはタスクを引き受けられません。潜在的なユーザーのロールを所有者除外リストに追加すると、そのユーザーはタスクを引き受けられません。

1. 【割り当て】 タブをクリックします。

次の図のように、【除外する所有者】 セクションが表示されます。

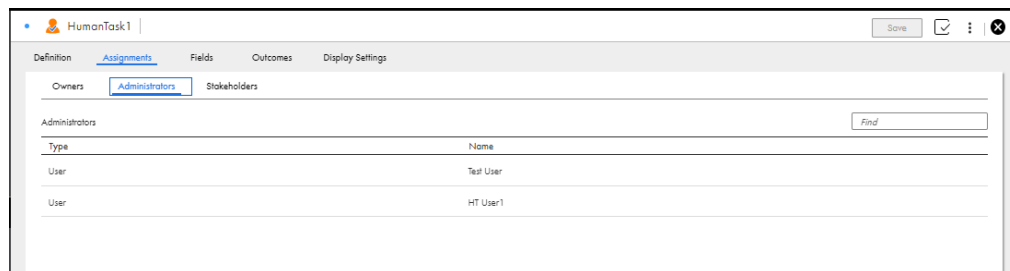


2. 【所有者】 タブで、【除外する所有者】 セクションの【+】をクリックし、リストからユーザーまたはロールを追加します。
3. 次のいずれかの手順を実行します。
  - ユーザーを追加するには、【ユーザー】をクリックして、リストからユーザーを選択します。【OK】をクリックします。
  - ロールを追加するには、【ロール】をクリックして、リストからロールを選択します。【OK】をクリックします。

## 手順 4.タスク管理者の定義

ヒューマンタスクアセットの【割り当て】 > 【管理者】 タブで、タスクを割り当ておよび再割り当てする権限を持つ管理者のリストを追加できます。

1. 【割り当て】 タブをクリックします。
2. 【管理者】 セクションをクリックします。次の図のように、【管理者】 セクションが表示されます。



3. 【+】をクリックして、リストからユーザーまたはロールを追加します。
4. 次のいずれかの手順を実行します。
  - ユーザーを追加するには、【ユーザー】をクリックして、リストからユーザーを選択します。【OK】をクリックします。

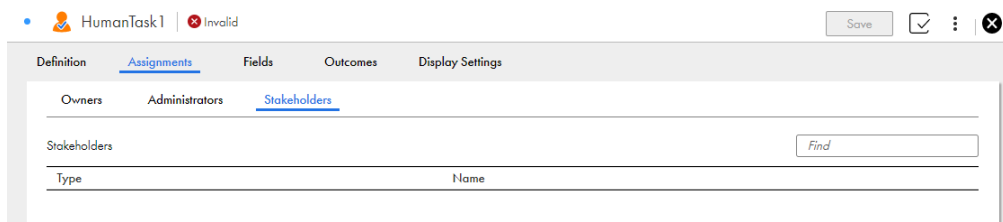
- ロールを追加するには、**【ロール】** をクリックして、リストからロールを選択します。**【OK】** をクリックします。

## 手順 5.関係者を定義する

ヒューマンタスクアセットの **【割り当て】** > **【関係者】** タブで、タスクを確認する権限を持つ関係者のリストを追加できます。問題が発生した場合、関係者は必要な措置を講じることができます。関係者はタスクにファイルを添付したり、コメントを追加したりすることもできます。タスクには複数の関係者を含めることができます。

1. **【割り当て】** > **【関係者】** タブをクリックします。

次の図のように、**【関係者】** タブが表示されます。



2. **【+】** をクリックして、リストからユーザーまたはロールを追加します。
3. 次のいずれかの手順を実行します。
  - ユーザーを追加するには、**【ユーザー】** をクリックして、リストからユーザーを選択します。**【OK】** をクリックします。
  - ロールを追加するには、**【ロール】** をクリックして、リストからロールを選択します。**【OK】** をクリックします。

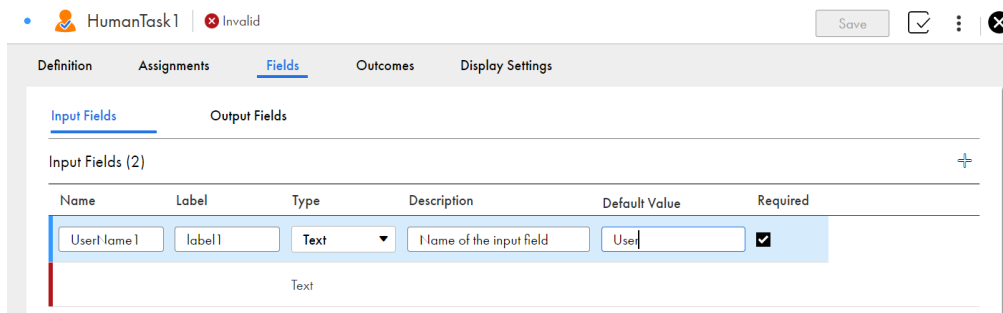
## 手順 6.入力フィールドと出力フィールドを設定する

ヒューマンタスクアセットの **【フィールド】** タブでは、フィールド名、フィールド型、フィールドの説明、デフォルト値など、すべての入力フィールドと出力フィールドのプロパティを指定できます。設定されたフィールドが必須かどうかも指定できます。

タスク所有者がヒューマンタスクに対処するために必要になる情報を表示する入力フィールドを設定します。タスク所有者がヒューマンタスクに関する情報を提供するために使用できる出力フィールドを設定します。

1. **【フィールド】** タブをクリックします。

次の図のように、**【入力フィールド】** セクションが表示されます。



2. 入力フィールドを追加するには、次のステップを実行します。
  - a. **[+]** をクリックして入力フィールドを追加します。
  - b. 入力フィールドの名前を入力します。
  - c. 入力フィールドのラベルを入力します。
  - d. リストからフィールド型を選択します。整数、チェックボックス、テキスト、日付、日時、数値、時刻、テキストエリア、およびオブジェクトを選択できます。タイプに **[オブジェクト]** を選択すると、プロセスオブジェクトを選択できます。プロセスオブジェクトは、情報をグループ化する場合に役立ちます。例えば、名前、住所、電話番号などの個別のフィールドを作成する代わりに、人口統計的なデータを含んださまざまなフィールドを持つ1つのプロセスオブジェクトを作成することができます。プロセスオブジェクトの作成の詳細については、「[プロセスオブジェクトのデザイン](#)」を参照してください。ヒューマンタスクアセットでプロセスオブジェクトを使用する際に考慮する必要があるルールとガイドラインについては、「[プロセスオブジェクトを使用するためのルールおよびガイドライン](#)」(ページ 286)を参照してください。
  - e. 必要に応じて、フィールドの説明を入力します。
  - f. **[デフォルト値]** フィールドに、そのフィールドで最もよく使用される値を入力するか、フィールドの設定方法に関するガイドラインテキストを入力します。
  - g. **[必須]** チェックボックスを選択して、タスクに必要な入力フィールドを設定します。
3. **[出力フィールド]** セクションをクリックします。

次の図のように、**[出力フィールド]** セクションが表示されます。

Name	Label	Type	Description	Default Value	Required
Output	output	Text	check	Enter a default value	<input type="checkbox"/>

4. 出力フィールドを追加するには、次の手順を実行します。
  - a. **[+]** をクリックして出力フィールドを追加します。
  - b. 出力フィールドの名前を入力します。
  - c. 出力フィールドのラベルを入力します。
  - d. リストからフィールド型を選択します。整数、チェックボックス、テキスト、日付、日時、数値、時刻、テキストエリア、およびオブジェクトを選択できます。タイプに **[オブジェクト]** を選択すると、プロセスオブジェクトを選択できます。プロセスオブジェクトは、情報をグループ化する場合に役立ちます。例えば、名前、住所、電話番号などの個別のフィールドを作成する代わりに、人口統計的なデータを含んださまざまなフィールドを持つ1つのプロセスオブジェクトを作成することができます。プロセスオブジェクトの作成の詳細については、「[プロセスオブジェクトのデザイン](#)」を参照してください。ヒューマンタスクアセットでプロセスオブジェクトを使用する際に考慮する必要があるルールとガイドラインについては、「[プロセスオブジェクトを使用するためのルールおよびガイドライン](#)」(ページ 286)を参照してください。
  - e. 必要に応じて、フィールドの説明を入力します。
  - f. **[デフォルト値]** フィールドに、そのフィールドで最もよく使用される値を入力するか、フィールドの設定方法に関するガイドラインテキストを入力します。

- g. **【必須】** チェックボックスを選択して、タスクに必要な出力フィールドを設定します。

## プロセスオブジェクトを使用するためのルールおよびガイドライン

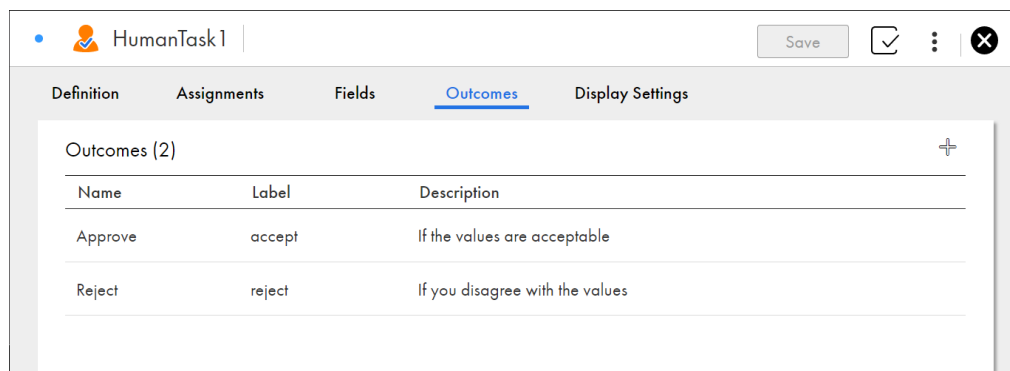
ヒューマンタスクアセットでプロセスオブジェクトを使用する場合は、次のルールとガイドラインを考慮してください。

- 入力フィールドと出力フィールドには、最大 10 個のプロセスオブジェクトを追加できます。
- 選択したプロセスオブジェクトには、最大 5 個のネストされたオブジェクトを含めることができます。
- 選択したプロセスオブジェクトには、ネストされたオブジェクトを含めて最大 50 個の入力フィールドと出力フィールドを含めることができます。
- 選択したプロセスオブジェクトに循環参照を含めることはできません。
- 選択したプロセスには、少なくとも 1 つのアイテムを含める必要があります。

## 手順 7.タスクの結果を設定する

ヒューマンタスクアセットの **【結果】** タブで、タスクの完了方法を指定できます。タスク所有者がタスクに対して下せる決定のリストを作成できます。例えば、タスクで考えられる結果として **【承認】** と **【却下】** を追加できます。

1. **【結果】** タブをクリックします。  
次の図は、**【結果】** タブを示しています。



2. **【+】** をクリックし、ヒューマンタスクサービスのヒューマンタスクインボックスに表示する必要がある結果を追加します。

## 手順 8.表示設定を指定する

ヒューマンタスクアセットの **【表示設定】** タブで、タスクがヒューマンタスクインボックスで使用する設定を定義します。タスクの表示名、タスクの件名、タスクの説明などのプロパティを指定できます。

ガイドレンダリングメソッドまたはデフォルトレンダリングメソッドを使用して、ヒューマンタスクインボックスにフィールドを表示できます。レンダリングメソッドとしてガイドを選択した場合、タスクはガイドで定義された画面を使用してフィールドを表示します。**【ガイド】** フィールドのレンダリングメソッドとしてガイドを選択しない場合、タスクはデフォルトレンダリングメソッドを使用してフィールドをヒューマンタスクインボックスに表示します。

ヒューマンタスクのガイドレンダリングメソッドを使用するには、2 つの方法があります。ガイドを選択するか、タスクがヒューマンタスクインボックスでのレンダリングに使用する必要がある新しいガイドを生成する

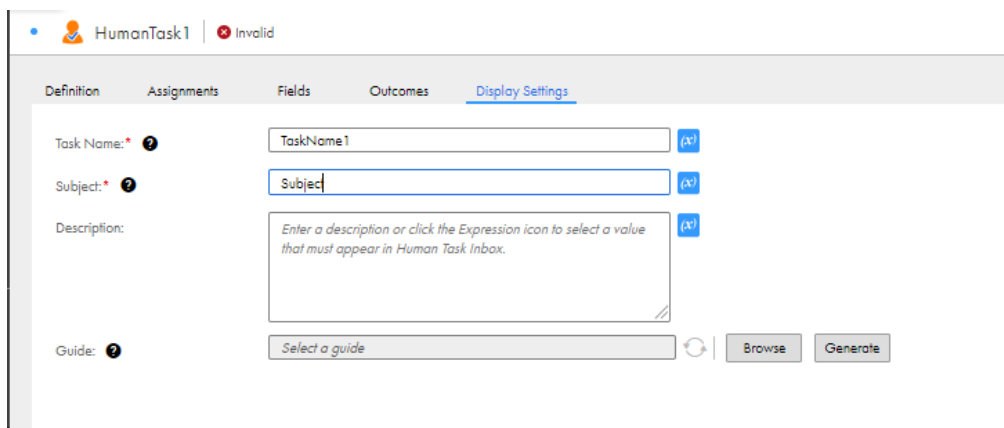
ことができます。**【参照】** をクリックすると、アプリケーション統合の既存のガイドを選択できます。ただし、入力フィールド、出力フィールド、選択したガイドの結果は、ヒューマンタスクアセットで指定する値に一致する必要があります。

**【生成】** をクリックすると、新しいガイドを作成できます。この機能を使用すると、アセットで定義した値に基づいて、入力フィールド、出力フィールド、および結果に事前に値が入力されたガイドを生成できます。さらにガイドを設定、保存、パブリッシュして、ヒューマンタスクで使用することができます。

**注:** 新しいガイドを生成する際にヒューマンタスクアセットの出力フィールドに必須フィールドを選択した場合でも、出力フィールドがガイドデザイナーに**【必須】** として表示されることはありません。必須の出力フィールドに値を指定しなくても、タスクを正常に完了し、プロセスの次の手順に進むことができます。

1. **【表示設定】** タブをクリックします。

次の図は、**【表示設定】** タブを示しています。



2. プロセスが生成するヒューマンタスクの名前を入力します。必要に応じて、**【タスク名】** フィールドの横にある **【式エディタ】** アイコンをクリックし、値を選択できます。
3. ヒューマンタスクインボックスに表示する必要があるタスクの件名を入力します。必要に応じて、**【件名】** フィールドの横にある **【式エディタ】** アイコンをクリックし、値を選択できます。
4. 必要に応じて、ヒューマンタスクインボックスに表示する必要があるタスクの説明を入力します。また、**【説明】** フィールドの横にある **【式エディタ】** アイコンをクリックして値を選択することもできます。
5. 必要に応じて、既存のガイドを選択するか、**【ガイド】** フィールドのヒューマンタスクにマッピングする新しいガイドを生成します。**【生成】** ボタンをクリックし、確認メッセージで **【OK】** をクリックして新しいガイドを生成します。
6. 要件に応じてガイドをさらに設定することができます。新しいガイドを生成した後に、ヒューマンタスクでそのガイドを使用するには、新しいガイドを保存してパブリッシュする必要があります。ガイドの生成中に発生したエラーは、ヒューマンタスクアセットに検証エラーとして表示されます。

## プロセス内でのヒューマンタスクの使用

ヒューマンタスクアセットを作成した後、人間の介入が必要なプロセスでそれを使用できます。

1. ヒューマンタスクステップをプロセスにドラッグし、作成したヒューマンタスクアセットを選択します。詳細については、[「ヒューマンタスクステップ」 \(ページ 77\)](#)を参照してください。
2. 必要に応じて、他のプロセスステップを設定します。

Process Designer で、または式を使用して値を設定するフィールドで、XQuery を操作するときにヒューマンタスクユーティリティ関数を使用することもできます。詳細については、「[ヒューマンタスク XQuery ユーティリティ](#)」 (ページ 50) を参照してください。

3. プロセスをクラウドサーバー上でパブリッシュして実行します。  
プロセスがヒューマンタスクステップを実行すると、ヒューマンタスクが生成され、ヒューマンタスクサービスのヒューマンタスクインボックスに送信されます。割り当てられたユーザーは、タスクを表示して対応できるようになります。

タスクが完了すると、プロセスはコールバックを受け取り、後続のステップに進みます。プロセスが完了すると、アプリケーション統合コンソールでヒューマンタスクステップの実行の詳細を確認できます。

ヒューマンタスクインボックスの詳細については、ヒューマンタスクのヘルプを参照してください。