



Adapter Development Kit による統合の設計

Service Link アダプタを開発するために、Service Link Adapter Development Kit (ADK) を使用できます。ADK は Service Catalog の Service Link サブシステム用のアダプタを作成できるコンポーネントセットです。Service Link は、Service Catalog に外部との通信を提供し、ワークフローを外部に公開して他のシステムと連携することができます。

この通信を実現するために、Service Link はインストール可能なアダプタをサポートしています。標準的なアダプタが Service Link に付属していますが、開発者が別途作成することも可能です。この章では、アダプタを作成するプロセスについて説明します。

この章の対象者は次のとおりです。

- **管理者。**管理者は、製品パッケージを使用できるため、サービス カタログ (Service Catalog) 製品をカスタマー システムにインストールできます。
- **アダプタ開発者。**アダプタ開発者とは、ANT を含む Java テクノロジーに精通した人であり、要求されている統合についての専門家です。

カスタム アダプタの作成に関するサポートについては、[Cisco TAC](#) に連絡してください。

使用する前に

ここでは、ADK のインストール、ADK の構造、アダプタのコンパイル、およびアダプタの導入について説明します。

JDK のインストール

Sun または IBM の説明に従って、Java Development Kit をインストールします。サービス カタログ (Service Catalog) で動作が保証されているのは、JBoss 7.1.1 のインストール環境では Sun JDK 6、WebSphere 7.0.0.17 のインストール環境では IBM Java 1.6 です。

ADK のインストール

ADK のインストール方法は次のとおりです。

1. **管理者**: 製品パッケージのコンテキストを展開し、image/isee/dist フォルダにある adk.zip を探し、その場所をアダプタ開発者に知らせます。
2. **アダプタ開発者**: ファイル adk.zip (または adk.tar.gz) を管理者から入手します。
3. **アダプタ開発者**: ADK パッケージをローカル マシン (C:\ADK) で展開します。C ドライブにする必要も、ADK ディレクトリにする必要もありません。ただし、この章の例では C:\ADK です。

ADK 構造

ADK をインストールすると、次のサブディレクトリが作成されます。

表 6-1 ADK のサブディレクトリ

| ディレクトリ | 説明 |
|----------------|---|
| <root> | ビルドプロシージャファイルが格納されます。 |
| ant | ANT ビルドシステム一式が格納されます。この ANT は標準の Apache ANT ビルドシステムであり、追加の拡張がいくつか含まれています。 |
| doc | java doc サブディレクトリが格納されます。ADK ドキュメントをここに配置できます。 |
| doc\javadoc | javadoc 形式の ADK のヘルプ。 |
| example | アダプタのサンプルが格納されます。 |
| example\src | サンプルアダプタのソースファイル Java ファイルが格納されます。 |
| example\deploy | そのアダプタについて記述された adapter.xml が格納されます。 |
| lib | コンパイルに必要な ADK ライブラリが格納されます。 |

アダプタは ADK 主構造のサブディレクトリです。インストール後の **example** も、そのようなアダプタの 1 つです。アダプタのコードは、次のような構造にする必要があります。

表 6-2 アダプタコード構造表

| ディレクトリ | 説明 |
|------------------|--|
| <adapter> | アダプタの短縮名。サンプルアダプタの場合は example です。 |
| <adapter>\src | 必須。ソース Java ファイルのルート。 |
| <adapter>\deploy | 必須。導入ディレクトリ。ここには、 adapter.xml というファイルが格納されている必要があります。 |
| <adapter>\lib | これはオプションです。 ISEE.war の lib ディレクトリに追加する、追加ライブラリ。 |
| <adapter>\config | これはオプションです。 ISEE.war のクラスディレクトリにコピーされる追加ファイル。 |

この例では、**src** および **deploy** だけです。

ファイルをコンパイルした後、ステージングディレクトリを作成します(アダプタの作成方法については、次の項を参照してください)。ステージングディレクトリは、作成手順を使用して後から削除および再作成できます。

表 6-3 ディレクトリの説明表

| ディレクトリ | 説明 |
|------------------|--|
| staging | 実稼働ディレクトリのルート。 |
| staging\classes | アダプタのコンパイルされた Java クラス。 |
| staging\adapters | 各アダプタのビルド済み jar が格納されます。アダプタの名前は、 adapter_<adaptershortname>.jar となります。 |
| staging\config | 各アダプタの config サブディレクトリにあったファイル。 |
| staging\deploy | それぞれの deploy サブディレクトリのファイル名は、 <adaptershortname>.xml に変更されます。 |
| staging\lib | 各アダプタの lib サブディレクトリにあったファイル。 |
| staging\dist | 最後の isee.adapters の展開可能なファイル。 |

アダプタのソース構造の作成

新しいアダプタを作成するには:

-
- 手順 1 前述のようなディレクトリ構造を作成します。
 - 手順 2 ソースを作成し、構造内に配置します。
 - 手順 3 **adapter.xml** を作成し、**deploy** ディレクトリに配置します。詳細については、[adapter.xml 記述子について](#)を参照してください。
 - 手順 4 オプションで、追加のライブラリおよび追加のコンフィギュレーションファイルを追加します。
 - 手順 5 **build.properties** を変更し、使用するアダプタをアダプタの行に追加します。これにより、作成したディレクトリを検索するよう ANT が設定されます。
-

コンパイル手順では、このビルドをバージョン管理に追加することができ、後でインストール前にコンパイルできます。

アダプタのコンパイル

アダプタを作成した後、次を実行することによってアダプタをビルドします。

build.cmd (UNIX システムの場合は **./build.sh**)

最終的な生成物が、**staging/dist/isee.adapters** に生成されます。導入するときは、このファイルを管理者に渡してください。

アダプタの導入

アダプタを導入するには、管理者が次の手順を実行します。

-
- 手順 1 Service Link のカスタム アダプタ パッケージを取得します。これは、zip ファイル形式でなければなりません。
 - 手順 2 一時ディレクトリ (c:\temp\adapter など) にアダプタを解凍します。このディレクトリを、今後 <AH> と呼びます。
 - 手順 3 展開された「ISEE.war/WEB-INF/lib」ディレクトリに <AH>/adapters/<ADAPTER_NAME>.jar をコピーします。
 - 手順 4 展開された「ISEE.war/WEB-INF/lib」ディレクトリに <AH>/lib/* ファイル (存在する場合) をコピーします。
 - 手順 5 展開された「ISEE.war/WEB-INF/classes」ディレクトリに <AH>/config/* ファイル (存在する場合) をコピーします。
 - 手順 6 展開された「ISEE.war/WEB-INF/classes」ディレクトリに <AH>/udk/* ファイル (存在する場合) をコピーします。
 - 手順 7 カスタム アダプタが Adapter Development Kit を使用して内部的に作成されていない場合は、「<ServicePortal_Software_Dir>/adk」フォルダから adk.zip を入手します。ここで、<ServicePortal_Software_Dir> はサービス カタログ (Service Catalog) アプリケーションの解凍されたソフトウェア イメージです。adk.zip を c:\adk (Windows の場合) または /opt/adk (UNIX/Linux の場合) に解凍します。このディレクトリを、今後 <ADK> と呼びます。
 - 手順 8 JAVA_HOME 環境変数が環境にまだ設定されていない場合は、これを設定します。
 - 手順 9 コマンド ウィンドウを開き、<ADK>/lib フォルダにディレクトリを変更します。環境に応じて adapter_dbinstaller.sh または adapter_dbinstaller.cmd を実行します。--help または -? をアダプタ インストーラの引数として使用し、必要な入力引数のリストを表示します。Adapter Deployment Descriptor ファイルの指定を求められたら、<AH>/deploy ディレクトリの下に xml ファイル名をフルパスで入力します (/opt/<AH>/deploy/custom_adapter.xml など)。
 - 手順 10 手順 6 でインストールされた各 udk ファイルについて、integrationserver.properties ファイル内の「UDConfig」プロパティにファイル名を追加します。UDConfig プロパティはすべての udconfig ファイルのカンマ区切りリストです。アダプタの udconfig ファイルをこのリストに追加します。
 - 手順 11 Service Catalog および Service Link サーバを起動します。新しいアダプタがあることを確認します。
-

アダプタの実装について

アダプタは Service Link が外部システム (通常はサードパーティ システムと呼ばれます) に接続する手段です。アダプタは、3 つの部分で構成されます。

- 着信部分。着信アダプタと呼ばれます。
- 発信部分。発信アダプタと呼ばれます。
- エラーハンドラ

着信アダプタは、Service Catalog への着信通信を管理します。これは、システムに届く XML メッセージを処理します。受信アダプタには、ポーラーとリスナーの 2 種類があります。ポーラーは定期的に起動して、着信メッセージを探すスレッドです。これに対し、リスナーは待機し、外部イベントによって起動されます。ポーラーの例としては、メッセージを定期的にチェックする必要がある受信ファイルアダプタがあります。リスナーの例としては、HTTP XML イベントがポストされるまで待機する HTTP アダプタがあります。

発信アダプタは Service Catalog から発信される XML メッセージを管理します。発信アダプタのタイプは 1 つだけです。

エージェントは論理要素であり、これによりサービス設計者は、複雑なアダプタおよび接続プロパティをすべて把握する必要がなくなります。また、エージェントは着信アダプタと発信アダプタを定義します。着信アダプタは任意選択で、「自動完了 (Auto complete)」と指定できます。「自動完了 (Auto complete)」はワークフローの進行にサードパーティ システムからの応答を必要としないモードで、ほとんどの場合、電子メールベース システムのような、信頼できない、または一度確立したら後は調整されないプロトコルと関連づけられます。管理者は、サービス設計者が使用できるようにエージェントおよびエージェントとアダプタとの関連付けを設定します。

また、メッセージがサードパーティ システムに送られる前、またはサードパーティ システムから受信され、Service Link に配信された後で、XML 変換をメッセージに適用できます。

メッセージ システムは nsXML と呼ばれる一般的な XML 方言を使用します。これは、Service Link が処理または作成できる有効な XML を定義するスキーマです。nsXML は、現在 6 つの操作で構成されています。

- task-started: 発信
- task-cancelled: 発信
- take-action: 着信
- send-parameters: 着信
- add-comment: 着信

発信の場合、Service Link はこれらの操作を宛先に変換できます。同じことが着信メッセージでも行われ、XSL 変換によって外部形式を nsXML の方言に変換できます。

nsXML 操作の詳細については、[通信メッセージのコンテンツと構造の理解](#)を参照してください。

アダプタの種類

アダプタには次の 2 つのタイプがあります。

- 転送アダプタ

転送アダプタは HTTP、ファイル、JMS、または独自のネットワーク ソケット実装など、特定の転送に固有のものです。

- アプリケーションアダプタ

アプリケーションアダプタには転送要素がありますが、Remedy や Siebel などの特定のサードパーティ アプリケーションを使用したほうがよく理解できます。多くの場合、jar によりネイティブ API が提供されます。このバージョンの Service Link では、転送アダプタを拡張してアプリケーションアダプタを作成することはまだ不可能です。

エージェントは着信メッセージと発信メッセージに異なるアダプタを使用することができます。

アダプタのコンポーネント

java コードの他には、次のものでアダプタは構成されています。

- Jar ライブラリ (Remedy java API など)
- 静的コンフィギュレーション ファイル。一度導入されたテキスト ファイルの変更は推奨されません。
- 展開記述子。アダプタを記述する XML ファイル。

接続プロパティ

アダプタは、Service Link モジュールが管理者に公開している接続プロパティを、サードパーティシステムに接続するために公開することができます。これらは XML 展開記述子内に記述されており、その値は java コードによって取得し、完成度の高い API に渡すことができます。

例: ファイルアダプタの実装

ここでは、簡単なアダプタを実装する方法を示します。サンプルアダプタは、外部環境との通信を行うファイルアダプタです。

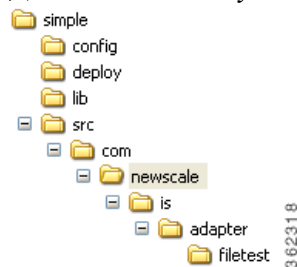
この単純なファイルアダプタには次のものが含まれています。

- ファイルを作成する発信アダプタ。ファイル名はアダプタのプロパティで指定されます。
- ファイルを読み取る着信アダプタ。ファイル名はアダプタのプロパティで指定されます。
- 単純な例外ハンドラ。

ディレクトリ構造の作成

最初に、アダプタのディレクトリ構造を作成します。ADK ディレクトリ構造内に、**simple** という名前のディレクトリを作成し、その下に次のディレクトリ構造を作成します。

図 6-1 Directory Structure



src の下が、java パッケージ **com.newscale.is.adapter.filetest** を表すソース パッケージになっていることに注目してください。他のパッケージも使用できますが、この例ではこれを使用します。

発信アダプタ クラスの作成

2 番目に、発信アダプタ クラスを作成します。名前は **FileOutboundAdapter** で、このクラス ファイルは、手順 1 で説明したパッケージに配置する必要があります。その枠組みを次に示します (メソッドの実装は含まれていません)。

```

import com.newscale.is.adk.AdapterContext;
import com.newscale.is.adk.base.OutboundAdapter;
import com.newscale.is.adk.exceptions.AdapterException;
public class FileOutboundAdapter extends OutboundAdapter {
    public FileOutboundAdapter (AdapterContext context) {
        super(context);
    }
    public void initiate (AdapterContext context) throws AdapterException {
    }
    public void processMessage (String message, String channelId) throws AdapterException {
    }
    public void terminate () throws AdapterException {
    }
    public void commit() throws AdapterException {
    }
    public void rollback() throws AdapterException {
    }
}

```

発信アダプタを作成するには、上記のようにクラス **com.newscale.is.adk.base.OutboundAdapter** をこのクラスで拡張する必要があります。

パラメータとして **com.newscale.is.adk.AdapterContext** を受け取るコンストラクタを実装します。このコンストラクタの実装に関して推奨される方法についても、上に示しています(スーパー コンストラクタを呼び出しています)。

上記のように **initiate** メソッドを実装します。このメソッドは、アダプタを使用するエージェントが起動されると呼び出されます。このメソッドが空の場合は省略可能です。

processMessage メソッドを実装します。このメソッドは、メッセージの送信準備が完了したときに呼び出されます。トランスフォーマがエージェントで指定されている場合、そのトランスフォーマによってメッセージが変換されています。

terminate メソッドを実装します。エージェントの停止時にこのメソッドを呼び出します。このメソッドが空の場合は省略可能です。

commit メソッドを実装します。エージェントがトランザクションを完了しようとしているときに、このメソッドを呼び出します。このメソッドが空の場合は省略可能です。このメソッドを使用すると、トランザクションを **processMessage** で開始し、後で完了できます。

rollback メソッドを実装します。このメソッドは、エージェントがトランザクションをロールバックしようとしているときに呼び出されます。このメソッドを空のままにする場合は省略可能です。このメソッドを使用すると、トランザクションを **processMessage** で開始し、後で再呼び出しできます。

トランザクションサポートの詳細については、[トランザクション通知の設定](#)を参照してください。

この例では、ファイル発信クラスは **xml** コンテンツを持つファイルを書き込みます。このために、ファイルが保存されファイル名を保持する変数を最初に設定します。これには、エージェントのプロパティを使用します。

```

String path = null;
public void initiate (AdapterContext context)
    throws AdapterException {
    Properties properties = context.getProperties();
    this.path = properties.getProperty("OB_FILE_DIR") + "/" +
        properties.getProperty("OB_FILE_NAME");
}

```

文字列が受信される場合、それをファイルに書き込むことは自明です。

```
public void processMessage (String message, String channelId)
    throws AdapterException {
    try {
        Writer w = new FileWriter(path);
        w.write(message);
        w.close();
    } catch (Exception e) {
        e.printStackTrace();
        throw new AdapterException(1, "Problem while writing to a file: " +
            e.getMessage());
    }
}
```

もちろん、このコードは説明のために大幅に簡素化しています。

ポーラー着信アダプタ クラスの作成

着信アダプタの枠組みは次のようになります。

```
public class FileInboundAdapter extends InboundAdapter {
    public FileInboundAdapter (AdapterContext context) {
        super(context);
    }
    public void initiate (AdapterContext context) throws AdapterException {
    }
    public String receiveMessage () throws AdapterException {
        return null;
    }
    public void terminate () throws AdapterException {
    }
    public void commit()
        throws AdapterException {
    }
    public void rollback()
        throws AdapterException {
    }
}
```

メソッドのセマンティクスは発信アダプタのものと同様です。唯一の例外は **receiveMessage** メソッドです。**receiveMessage** メソッドは、ポーラーアダプタの場合は定期的呼び出されます。データが検出されると、このメソッドは有効な xml をサードパーティ形式で返します。データが検出されなかった場合は、ヌルが返されます。着信アダプタのコードは次のとおりです(発信アダプタと同様、正しいパラメータを使用して初期化が行われます)。

```
String path = null;
public void initiate (AdapterContext context)
    throws AdapterException {
    Properties properties = context.getProperties();
    this.path = properties.getProperty("IB_FILE_DIR") + "/" +
        properties.getProperty("IB_FILE_NAME");
}
```

ファイルの処理は次のように行われます。

```
public String receiveMessage () throws AdapterException {
    String receivedMessage = "";
    char data[] = {};
    try {
        StringBuffer buffer = new StringBuffer();
        FileInputStream fis = new FileInputStream(path);
        InputStreamReader isr = new InputStreamReader(fis, "UTF8");
```



```

    Reader in = new BufferedReader(isr);
    int ch;
    while ((ch = in.read()) > -1) {
        buffer.append((char) ch);
    }
    in.close();
    String requestString = buffer.toString();
    boolean success = (new File(path)).delete();
    return requestString;
} catch (Exception e) {
    return null;
}
}
}

```

リスナー着信アダプタの作成

リスナー アダプタは、Ad-Hoc プロセスを利用して作成されます。着信アダプタ クラスと、servlet のような実際のレシーバクラスという2つのクラスが動作します。レシーバクラスは、チャンネル ID を取得する必要があります。レシーバクラスは、次のようにして **InboundAdapter** クラスを発見します。

```

ChannelInfoVO chVo = AgentDAO.getInstance().getChannelInfo(channelId);
if(chVo != null){
    Adapter adapter =
        AgentManager.getInstance().getAdapter(chVo.getAgentId());
    ((InboundAdapter).receiverProcess(xml);
}
}

```

着信アダプタには、メッセージにより呼び出す必要がある **receiverProcess** メソッドか、または、**toString()** メソッドがメッセージのテキストを返すオブジェクトがあります。サンプルには、リスナー着信アダプタは用意されていません。

例外ハンドラの実装

2つのアダプタが完成したら、例外ハンドラを実装する必要があります。ここでは非常に単純なクラスを使用しているため、必要な処理はエラーの出力のみです。完全なクラスを以下に示します。

```

public class FileExceptionHandler implements ITransportExceptionHandler {
    public FileExceptionHandler () {
    }
    public void handleException (Map props, String message) {
        System.out.println("Outbound Message Failed to deliver: " + message);
    }
}
}

```

トランザクション通知の設定

アダプタにはトランザクション サポートが提供されているため、アダプタのトランザクションを取り消す前にはエージェントへの通知が行われます。この目的のために、メソッド **commit** および **rollback** が追加されています。



(注)

システムでトランザクションのコミットまたはロールバックを実行中に、これらのメソッドに論理コードを追加すべきではありません。これらのメソッドでロールバックまたはコミットするのは、自身のリソースのみにする必要があります。

次の一般的な手法をアダプタで使用すると、コミットまたはロールバックされたリソースをトラッキングすることができます。

静的マップを作成します。処理メソッド(`processMessage` または `receiveMessage`)が呼び出されると、そのメソッドは次のものを追加できます。

```
private static Map resources = new HashMap();
public void processMessage (String message, String channelId)
    throws AdapterException {
    Connection con = ... // obtain a connection to external resource
    Map.put (Thread.currentThread(), con);
}

public void commit() throws AdapterException {
    con = (Connection) map.get (Thread.currentThread());
    map.remove (Thread.currentThread());
    con.commit();
}
```

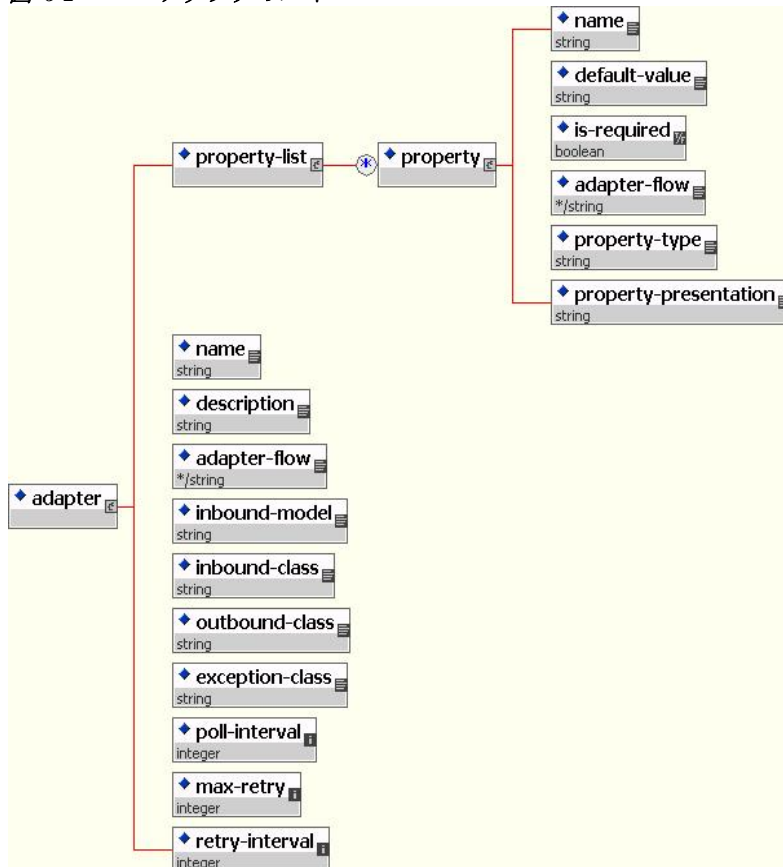
adapter.xml 記述子について

アダプタ記述子には、アダプタの導入とプロパティに関する情報が格納されます。

アダプタのスキーマ

アダプタのスキーマを次に示します。

図 6-2 アダプタのスキーマ



362320

「adapter」要素のフィールドの説明

name: アダプタの名前

description: アダプタの説明

adapter-flow:

有効な値は次のとおりです。

- 「inbound」
- 「outbound」

inbound-model:

有効な値は次のとおりです。

- 「listener」
- 「poller」
- 「extendedpoller」

inbound-class: 着信アダプタの絶対クラス名

outbound-class: 発信アダプタの絶対クラス名

exception-class: このアダプタの例外ハンドラの絶対クラス名

poll-interval: ミリ秒単位のポーリング間隔(「poller」タイプのアダプタに適用される)

max-retry: メッセージが失敗した場合に行われる再試行の最大回数

retry-interval: 再試行間隔(ミリ秒)

「property」(アダプタ プロパティ)要素のフィールドの説明。

name: アダプタ プロパティの名前

default-value: プロパティのデフォルト値

is-required: 必須プロパティとオプションプロパティのいずれであるか。有効な値は「true」または「false」

property-type: プロパティの種類。現在のところ、有効な値は「string」です。

property-presentation: 有効な値は「text」および「password」

adapter-flow:

有効な値は次のとおりです。

- 「inbound」
- 「outbound」

Adapter.xml の例

```
<?xml version="1.0" encoding="UTF-8"?>
<adapter>
<property-list>
<property>
<name>InboundFinalResolution</name>
<default-value>Preserve</default-value>
<is-required>true</is-required>
<adapter-flow>inbound</adapter-flow>
<property-type>string </property-type>
```

```

<property-presentation>text</property- presentation>
</property>
<property>
<name>InboundFileLocation</name>
<default-value>C://SL2//InboundFiles</default-value>
<is-required>>true</is-required>
<adapter-flow>inbound</adapter-flow>
<property-type>string </property-type>
<property-presentation>text</property- presentation>
</property>
<property>
<name>OnError</name>
<default-value>Preserve</default-value>
<is-required>>true</is-required>
<adapter-flow>inbound</adapter-flow>
<property-type>string </property-type>
<property-presentation>text</property- presentation>
</property>
<property>
<name>InboundBackupLocation</name>
<default-value>C://SL2//InboundBackup</default-value>
<is-required>>true</is-required>
<adapter-flow>inbound</adapter-flow>
<property-type>string </property-type>
<property-presentation>text</property- presentation>
</property>
<property>
<name>BackupSuffix</name>
<default-value>.bak</default-value>
<is-required>>true</is-required>
<adapter-flow>inbound</adapter-flow>
<property-type>string </property-type>
<property-presentation>text</property- presentation>
</property>
<property>
<name>FileNameDateFormat</name>
<default-value>.yyyyMMddHHmmssSSS</default-value>
<is-required>>true</is-required>
<adapter-flow>inbound</adapter-flow>
<property-type>string </property-type>
<property-presentation>text</property- presentation>
</property>
<property>
<name>InboundTempLocation</name>
<default-value>C://SL2//InboundTemp</default-value>
<is-required>>true</is-required>
<adapter-flow>inbound</adapter-flow>
<property-type>string </property-type>
<property-presentation>text</property- presentation>
</property>
<property>
<name>OutboundConflictResolution</name>
<default-value>Rename</default-value>
<is-required>>true</is-required>
<adapter-flow>outbound</adapter-flow>
<property-type>string </property-type>
<property-presentation>text</property- presentation>
</property>
<property>
<name>OutboundFileLocation</name>
<default-value>C://SL2//InboundFiles</default-value>
<is-required>>true</is-required>
<adapter-flow>outbound</adapter-flow>
<property-type>string </property-type>

```

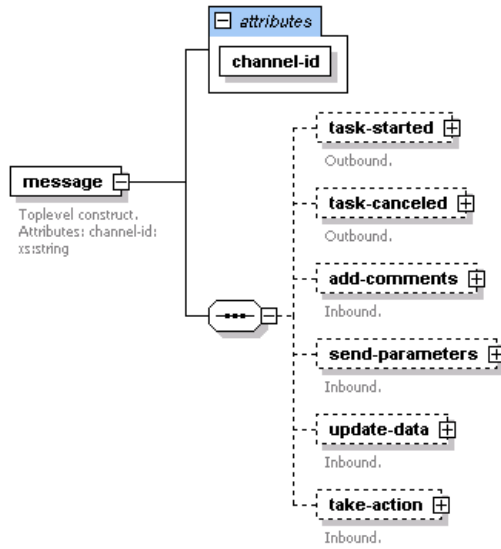
```
<property-presentation>text</property- presentation>
</property>
<property>
<name>OutboundBackupLocation</name>
<default-value>c://SL2//InboundBackup</default-value>
<is-required>>true</is-required>
<adapter-flow>outbound</adapter-flow>
<property-type>string </property-type>
<property-presentation>text</property- presentation>
</property>
<property>
<name>OutboundTempLocation</name>
<default-value>C://SL2//InboundTemp</default-value>
<is-required>>true</is-required>
<adapter-flow>outbound</adapter-flow>
<property-type>string </property-type>
<property-presentation>text</property- presentation>
</property>
</property-list>
<name>File Adapter</name>
<description>Read/write the external data from/to external file system</description>
<adapter-flow>inbound</adapter-flow>
<inbound-model>poller</inbound-model>
<inbound-class>com.newscale.is.adapter.file.FileInboundAdapter</inbound-class>
<outbound-class>com.newscale.is.adapter.file.FileOutboundAdapter</outbound-class>
<exception-class>com.newscale.is.adapter.file.FileExceptionHandler</exception-class>
<poll-interval>10000</poll-interval>
<max-retry>0</max-retry>
<retry-interval>0</retry-interval>
</adapter>
```

通信メッセージのコンテンツと構造の理解

ここでは、通信メッセージの内容と構造について詳しく説明します。メッセージの内容は、Service Catalog とサードパーティ システムの間で、HTTP、SOAP、JMS などの多様なキャリア プロトコルによって送信される XML 文書内にカプセル化されます。メッセージの構造およびサブ構造を理解しやすいよう、図に示します。

メッセージ(Message)

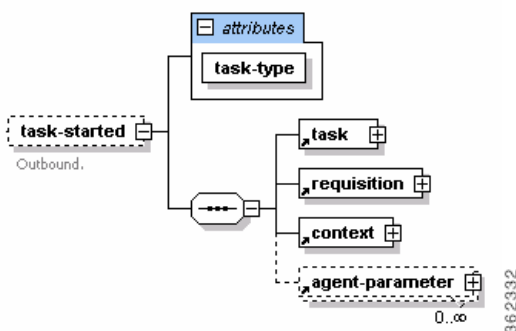
図 6-3 メッセージ(Message)



発信文書のトップレベル要素 `message` には、「`task-started`」または「`task-canceled`」要素が格納されます。着信文書のトップレベル要素 `message` には、「`add-comments`」、「`send-parameters`」、または「`take-action`」要素が1つ以上格納されます。`message` タグには、`channel-id` という文字列タイプの必須属性があります。これは、作成された発信メッセージに対して Service Link が生成する一意の文字列値です。サードパーティシステムは、メッセージに対応する `channel-id` で応答する必要があります。この ID は、サービスカタログ (Service Catalog) とサードパーティシステムの両方で引き継がれる必要があります。

Task Started または Task Cancelled

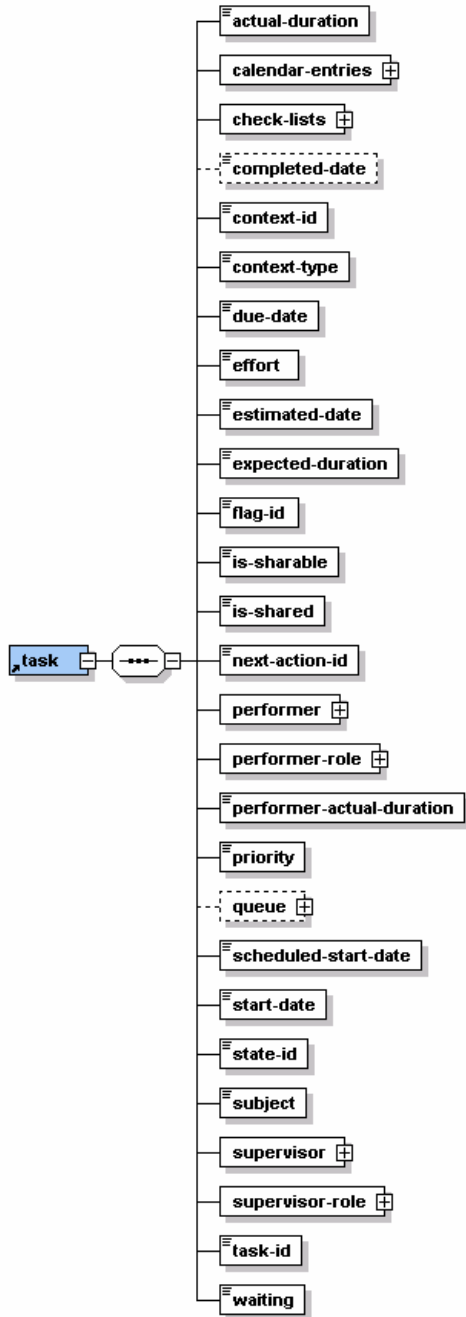
図 6-4 Task Started または Task Cancelled



Task started は、サードパーティシステムで外部アクティビティを開始します。この操作に関する設計上のポイントは、サードパーティシステムでタスクを実行するために必要となる可能性がある情報をすべて組み込むことです。この要素には、タスクおよびそのタスクが属している要求に関する、必要な詳細情報がすべて保持されます。また、オプションのエージェントパラメータも1つ以上含めることができます。`context` 要素は、サービス配信計画のコンテキスト内のタスクについて記述したものです。このノードに、要求レベルの確認および許可に関する値は格納されません。

タスク

図 6-5 タスク



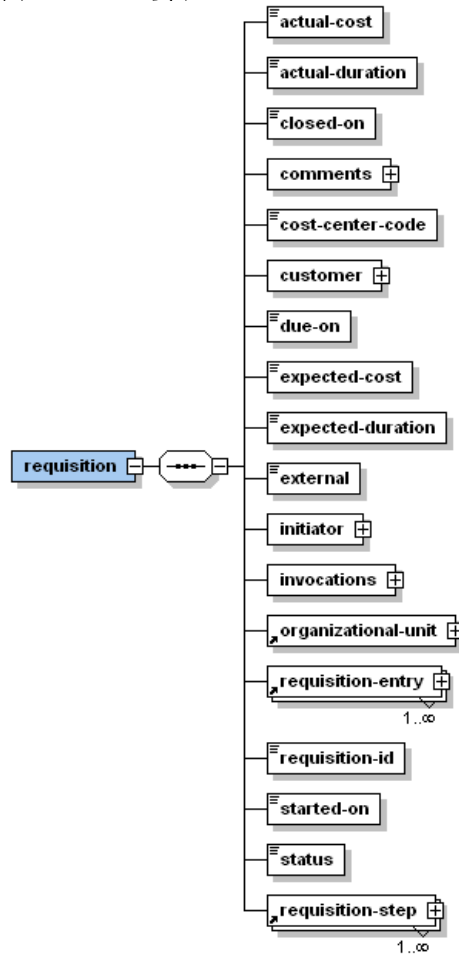
362333

表 6-4 タスク要素と説明

| 要素 | 説明 |
|---------------------------|---|
| actual-duration | タスクは実行中になったばかりであるため、空です。 |
| calendar-entries | サービスを要求した個人のカレンダー エントリ。 |
| check-lists | タスクのチェック リスト。 |
| completed-date | タスクはまだ完了していないため、空です。 |
| context-id | タスクが開始されたコンテキスト オブジェクトのオブジェクト ID。 |
| context-type | オブジェクトのコンテキスト(例: 要求エントリ)。 |
| due-date | タスクの期日。 |
| 実行 | 予想されるタスクの時間単位の工数(1人でタスクを完了するために必要な作業時間の予想)。 |
| estimated-date | 予測されるタスクの完了日時。 |
| expected-duration | 予想されるタスクの時間単位の所要期間(タスクの開始から完了までに要する作業時間の予想)。 |
| flag-id | UI のカラー インジケータ。 |
| is-sharable | タスクが共有可能かどうかを示すブール値。 |
| is-shared | タスクが共有されているかどうかを示すブール値。 |
| next-action-id | タスクに関して次に実行可能なアクションは何か。 |
| performer | タスクの実行者。performer 要素には、関連付けられた個人オブジェクトがあります。 |
| performer-actual-duration | タスクはまだ完了されていないため、空です。 |
| performer-role | 実行者が果たしている処理ロールは何か。 |
| [プライオリティ (priority)] | タスクの優先順位: 1、2、または 3(それぞれ高、中、または低を表す)。 |
| キュー | タスクが割り当てられているキューの説明。 |
| scheduled-start-date | タスクの開始が予定された日付。 |
| start-date | タスクが開始された日付。 |
| state-id | タスクの状態。 |
| subject | タスクの件名。件名は、サービス定義で変更されます。要求エントリでは変更されません。 |
| スーパーバイザ | タスクの上司。supervisor 要素には、関連付けられた個人オブジェクトがあります。 |
| supervisor-role | 上司が果たしている処理ロール。 |
| task-id | このタスク インスタンスを一意に識別するために使用される整数。要求エントリのタスクごとに新しい番号が生成されます。 |
| 待機中 | サブタスクおよびサードパーティ システムを含む、このタスクの依存関係を表すインジケータ。 |

要求

図 6-6 要求



requisition 要素は、すべての要求と要求エン트리 データをカプセル化したものであり、外部アクティビティを実行するときに、統合の目的でこのデータを使用できます。

表 6-5 要求要素の説明表

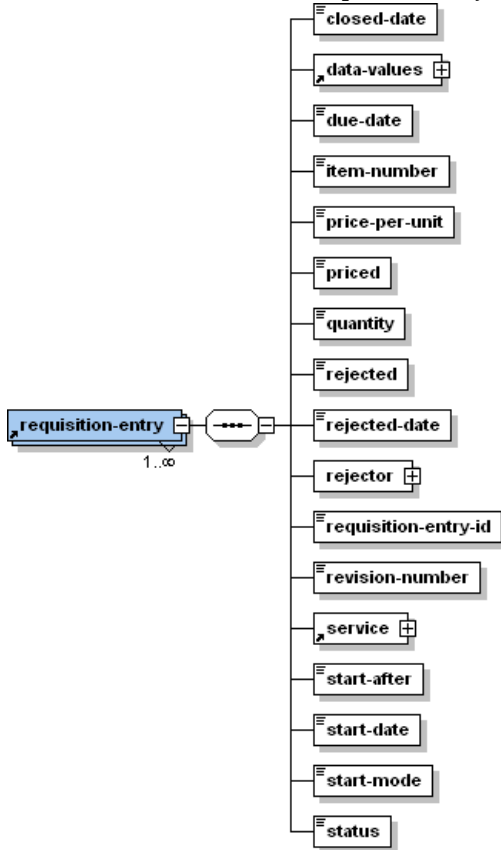
| 要素 | 説明 |
|------------------|---|
| サービス | 要求されるサービス数(または要求エン트리数)。 |
| actual-cost | 要求の実際のコスト。 |
| actual-duration | 要求の実際の所要期間。 |
| closed-on | 要求はまだ完了していないため、空です。 |
| コメント | 要求に関するコメント。 |
| cost-center-code | 未使用。 |
| インタビューの | 要求が注文された目的となっている個人。個人のオブジェクトデータが保持されます。 |
| due-on | 要求の配信期限を表す日時。 |

表 6-5 要求要素の説明表 (続き)

| 要素 | 説明 |
|---------------------|---|
| expected-cost | 予想される要求のコスト。 |
| expected-duration | 要求全体の処理にかかる予想される所要期間 (時間単位)。 |
| 外部 | 要求が外部システムから開始されたかどうかを示すブール値。 |
| initiator | この要求をオーダーした個人。個人のオブジェクト データが保持されます。 |
| 呼び出し | RAPI (要求 API) からセットアップされた属性。 |
| organizational-unit | 要求者 (発信者) の組織単位。 |
| requisition-entry | 要求エントリのデータ。 |
| requisition-id | 送信された要求の整数 ID。要求を手動で送信した後に My Services および Service Manager に表示される ID と同じです。 |
| requisition-step | 要求の承認または配信のステップとステータス。 |
| started-on | 要求が開始された日付。 |
| status | 要求の現在の状態。外部アクティビティの実行中は Ongoing となります。 |

要求エントリ

図 6-7 要求エントリ (Requisition Entry)



3 62354

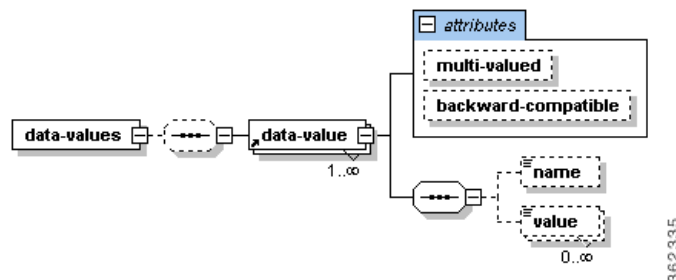
このタグは、1つの要求エントリのデータをすべてカプセル化したものであり、統合のために使用できます。

表 6-6 要求エントリ要素の説明表

| 要素 | 説明 |
|----------------------|--|
| closed-date | 要求エントリのステータスが ongoing から completed に変化した日時。タスクが ongoing である間は要求がクローズされないため、空になります。 |
| data-values | 要求エントリのデータ値。 |
| due-date | 要求が終了する予定の日付。 |
| item-number | 要求内の要求エントリの項目番号。 |
| price-per-unit | 要求されたサービスの単価。 |
| priced | 要求の価格が設定されている場合は true、設定されていない場合は false。 |
| quantity | 注文された数量。 |
| rejected | 要求エントリが承認されたか、拒否されたかを示します。 |
| rejected-date | 拒否された場合は、その日付。 |
| rejector | 要求を拒否した個人を示します。 |
| requisition-entry-id | エントリ ID。 |
| revision-number | リビジョンが作成されている場合は、そのリビジョン番号を示します。 |
| service | 要求エントリが属しているサービスに関連する要素。 |
| start-after | 遅れて開始される日付。 |
| start-date | 要求エントリが開始された日付。 |
| start-mode | 要求エントリがすぐに開始されるか、遅延されるかを指定します。 |
| status | 要求エントリのステータス: closed または ongoing。 |

データ値

図 6-8 データ値



data-values 要素は、ディクショナリ データで構成されるデータ値を 1 つ以上持つことができます。data-value の name は「Dictionaryname.FieldName」を指し、value はサービスの注文時にユーザが入力した値です。その値が複数選択ドロップダウン リストになっている場合は、1 つの data-value 要素が複数の値を持つことができます。

サービス

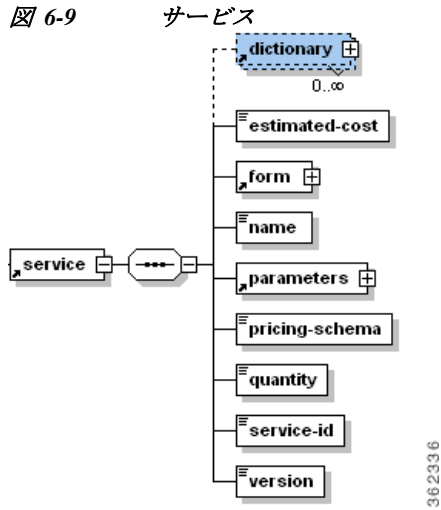


表 6-7 サービス要素の説明表

| 要素 | 説明 |
|-------------------|---|
| ディクショナリ | service 要素は、ゼロ個以上のディクショナリを持つことができます。 |
| estimated-cost | 予測されるサービスのコスト。 |
| form | サービス フォームのすべてのフィールド要素を保持する要素。 |
| 名前 | サービスの名前。 |
| パラメータ | このサービスに定義されているパラメータ。 |
| pricing-schema | サービスが、入札で価格設定を行うタスクであるか、固定価格であるかを指定します。 |
| quantity | 注文された数量。 |
| service-id | Service Catalog のサービスの ID。 |
| version | サービスの最終更新バージョン番号。 |
| standard-duration | サービスの標準的な所要期間。 |

辞書 (Dictionary)

図 6-10 デクショナリ

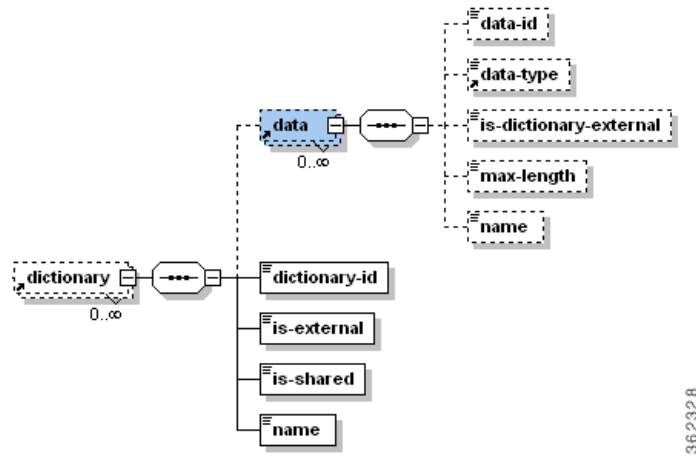
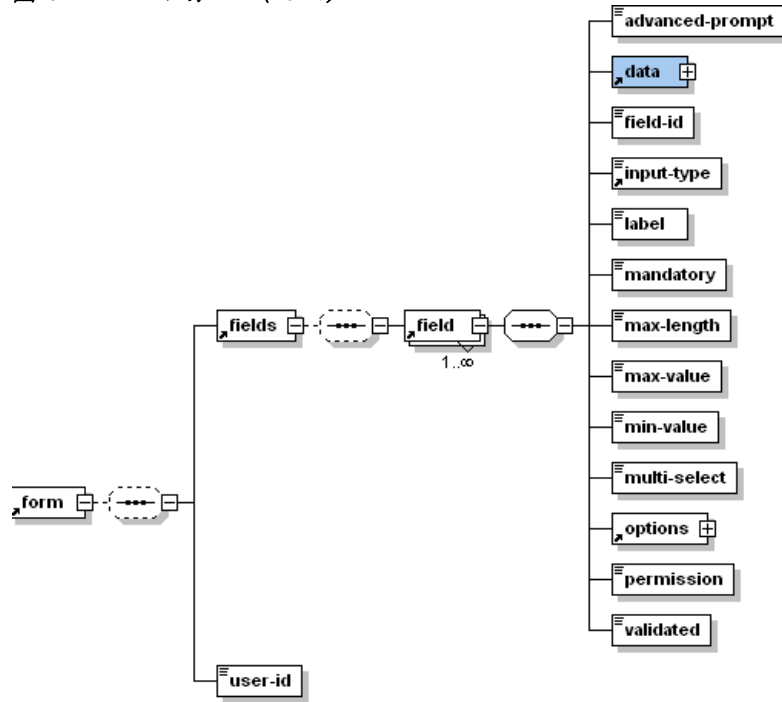


表 6-8 デクショナリ要素の説明表

| 要素 | 説明 |
|-----------------------------|---|
| caption | デクショナリ内のキャプション データが格納された文字列値。 |
| データ | デクショナリ内のデータ要素。データ要素には、データ要素の名前、最大長、データ型、およびその他のメタデータの値が保持されます。 |
| dictionary-id | Service Catalog 内のデクショナリのデクショナリ ID。 |
| dictionary-template-type-id | デクショナリの作成に使用されたテンプレート(たとえば、個人ベースのデクショナリの場合は 2)。 |
| classification-id | デクショナリのカテゴリ(サービス項目デクショナリの場合のみ)。 |
| mdr-data-type-id | デクショナリのサービス項目タイプ(サービス項目デクショナリの場合のみ)。 |
| display-order | デクショナリが表示順序が入った整数値。 |
| is-external | デクショナリが内部 Service Catalog デクショナリであるか、外部デクショナリであるかを示すブール値。 |
| is-reportable | デクショナリが、Advanced Reporting モジュールの Ad-Hoc レポート機能での使用に関して、レポート可能としてマークされているかどうかを示すブール値。 |
| is-shared | デクショナリが共有デクショナリであるかどうかを示すブール値。 |
| is-template | デクショナリがテンプレートであるかどうかを示すブール値。この値は必ず false になります。 |
| logic-name | デクショナリの内部名(予約デクショナリの場合のみ)。 |
| 名前 | デクショナリの名前。 |

フォーム

図 6-11 フォーム (Form)



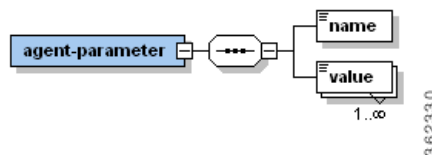
362329

表 6-9 フォーム要素の説明表

| 要素 | 説明 |
|---------|--|
| フィールド | <p>フィールドは、要求フォームの内部に 1 つまたは複数の field 要素を持ちます。</p> <p>advanced-prompt: リッチ HTML プロンプト。</p> <p>data: データ型、名前、長さなどのフィールドのデータが保持されます。</p> <p>dictionary-display-order: dictionary-display-order の値は、フィールドに関連付けられた DataElement に関連付けられている Dictionary の DefObjectDictionaries.DisplayOrder の値です。</p> <p>display-order: display-order の値は、フィールドの DefObjectDataHTML.DisplayOrder の値です。</p> <p>field-id: Service Catalog データベース内でのフィールド ID。</p> <p>input-type: フィールドの入力の種類 (例: text, option など)。</p> <p>label: フィールドに指定されているラベル。</p> <p>mandatory: フィールドのデータは、サービスで必須です。</p> <p>max-length: フィールドに指定されている最大長。</p> <p>max-value: 数値の場合は範囲が指定されます。</p> <p>min-value</p> <p>multi-select: 入力の種類が複数選択ボックスかどうか。</p> <p>options: このデータ フィールドに使用可能なオプション リスト。</p> <p>permission:</p> <p>validated: 検証が必要かどうか。</p> |
| user-id | |

エージェント パラメータ

図 6-12 エージェント パラメータ



エージェント パラメータは、エージェントに対して指定されている外部パラメータを表します。これには **multi-valued** というブール属性があり、ユーザが複数の値を選択できるパラメータかどうかに基づいて **true** または **false** のいずれかになります。**name** はエージェント パラメータの名前を表し、**value** はその値を表します。

着信および発信ドキュメントの例

task-started または task-canceled (発信)

```
<?xml version="1.0" encoding="UTF-8"?>
<message channel-id="18071221:1124919814742:-32752"
xmlns:fo="http://www.w3.org/1999/XSL/Format">
  <task-started task-type="task">
    <task>
      <actual-duration>0.0</actual-duration>
      <calendar-entries>
        <calendar-entry>
          <calendar-entry-id>2</calendar-entry-id>
          <date>Thu Aug 25 17:00:00 PDT 2005</date>
          <end-time>Fri Aug 26 21:40:37 PDT 2005</end-time>
          <is-blocked>false</is-blocked>
          <is-break>false</is-break>
          <is-read>false</is-read>
          <person>
            <company-address/>
            <email>admin@company.com</email>
            <fax/>
            <first-name>admin</first-name>
            <home-ou>
              <name>&lt;s ID=&quot;847&quot; /&gt;</name>
              <organizational-unit-id>1</organizational-unit-id>
            </home-ou>
            <home-phone/>
            <last-name/>
            <login-name>admin</login-name>
            <person-id>1</person-id>
            <personal-address/>
            <supervisor-name/>
            <timezone>Pacific Standard Time</timezone>
            <work-phone/>
          </person>
          <sequence>0</sequence>
          <start-time>Thu Aug 25 21:40:37 PDT 2005</start-time>
          <subject>External Task</subject>
        </calendar-entry>
      </calendar-entries>
      <check-lists>
        <check-list-entry>
          <display-order>1</display-order>
          <is-mandatory>true</is-mandatory>
          <last-date/>
          <last-person/>
          <name>Make sure you wake up</name>
          <status>false</status>
        </check-list-entry>
        <check-list-entry>
          <display-order>2</display-order>
          <is-mandatory>true</is-mandatory>
          <last-date/>
          <last-person/>
          <name>Make sure you take a shower</name>
          <status>false</status>
        </check-list-entry>
        <check-list-entry>
          <display-order>3</display-order>
```



```

        <is-mandatory>true</is-mandatory>
        <last-date/>
        <last-person/>
        <name>Make sure you have breakfast</name>
        <status>>false</status>
    </check-list-entry>
</check-lists>
<completed-date/>
<context-id>1</context-id>
<context-type>Requisition Entry</context-type>
<due-date>Fri Aug 26 21:40:37 PDT 2005</due-date>
<effort>10.0</effort>
<estimated-date/>
<expected-duration>10.0</expected-duration>
<flag-id>0</flag-id>
<is-sharable>>false</is-sharable>
<is-shared>>false</is-shared>
<next-action-id>2</next-action-id>
<performer>
    <company-address/>
    <email>admin@company.com</email>
    <fax/>
    <first-name>admin</first-name>
    <home-ou>
        <name>&lt;s ID=&quot;847&quot;/&gt;</name>
        <organizational-unit-id>1</organizational-unit-id>
    </home-ou>
    <home-phone/>
    <last-name/>
    <login-name>admin</login-name>
    <person-id>1</person-id>
    <personal-address/>
    <supervisor-name/>
    <timezone>Pacific Standard Time</timezone>
    <work-phone/>
</performer>
<performer-actual-duration>0.0</performer-actual-duration>
<priority>2</priority>
<scheduled-start-date>Thu Aug 25 21:40:37 PDT 2005</scheduled-start-date>
<start-date>Wed Aug 24 21:42:15 PDT 2005</start-date>
<state-id>2</state-id>
<subject>External Task</subject>
<supervisor>
    <company-address>Foo Bar 25 Suite 300 Foo City CA 94404
USA</company-address>
    <email>internal@company.com</email>
    <fax/>
    <first-name>Monkey</first-name>
    <home-ou>
        <name>&lt;s ID=&quot;847&quot;/&gt;</name>
        <organizational-unit-id>1</organizational-unit-id>
    </home-ou>
    <home-phone/>
    <last-name>McBride</last-name>
    <login-name>monkey</login-name>
    <person-id>3</person-id>
    <personal-address>Fuchi Caca 16 Apartment C Fuchi Minn OR 78787
USA</personal-address>
    <supervisor-name/>
    <timezone>Pacific Standard Time</timezone>
    <work-phone/>
</supervisor>
<task-id>3</task-id>
<waiting>1</waiting>

```

```

</task>
<requisition>
  <actual-cost>0.0</actual-cost>
  <actual-duration>0.0</actual-duration>
  <closed-on/>
  <comments>
    <comment>
      <comment-date>Wed Aug 24 21:42:06 PDT 2005</comment-date>
      <comment-id>1</comment-id>
      <comment-text>I am adding a comment and I cannot think of a better
comment</comment-text>
      <component-id>3</component-id>
      <component-name>Request Center Component</component-name>
      <person>
        <company-address/>
        <email>admin@company.com</email>
        <fax/>
        <first-name>admin</first-name>
        <home-ou>
          <name>&lt;s ID=&quot;847&quot;&gt;&/name>
          <organizational-unit-id>1</organizational-unit-id>
        </home-ou>
        <home-phone/>
        <last-name/>
        <login-name>admin</login-name>
        <person-id>1</person-id>
        <personal-address/>
        <supervisor-name/>
        <timezone>Pacific Standard Time</timezone>
        <work-phone/>
      </person>
      <source-object-id>2</source-object-id>
      <source-object-inst-id>1</source-object-inst-id>
    </comment>
  </comments>
  <cost-center-code/>
  <customer>
    <company-address/>
    <email>admin@company.com</email>
    <fax/>
    <first-name>admin</first-name>
    <home-ou>
      <name>&lt;s ID=&quot;847&quot;&gt;&/name>
      <organizational-unit-id>1</organizational-unit-id>
    </home-ou>
    <home-phone/>
    <last-name/>
    <login-name>admin</login-name>
    <person-id>1</person-id>
    <personal-address/>
    <supervisor-name/>
    <timezone>Pacific Standard Time</timezone>
    <work-phone/>
  </customer>
  <due-on>Fri Aug 26 21:40:37 PDT 2005</due-on>
  <expected-cost>0.0</expected-cost>
  <expected-duration>0.0</expected-duration>
  <external>false</external>
  <initiator>
    <company-address/>
    <email>admin@company.com</email>
    <fax/>
    <first-name>admin</first-name>
    <home-ou>

```

```

        <name>&lt;s ID=&quot;847&quot;/&gt;</name>
        <organizational-unit-id>1</organizational-unit-id>
    </home-ou>
    <home-phone/>
    <last-name/>
    <login-name>admin</login-name>
    <person-id>1</person-id>
    <personal-address/>
    <supervisor-name/>
    <timezone>Pacific Standard Time</timezone>
    <work-phone/>
</initiator>
<invocations/>
<organizational-unit>
    <name>&lt;s ID=&quot;847&quot;/&gt;</name>
    <organizational-unit-id>1</organizational-unit-id>
</organizational-unit>
<requisition-entry>
    <closed-date/>
    <data-values>
        <data-value>
            <name>Requester</name>
            <value>John McGarzafi</value>
        </data-value>
        <data-value>
            <name>RemedyStuff.TicketID</name>
            <value>None yet</value>
        </data-value>
        <data-value>
            <name>RemedyStuff.AssetNumber</name>
            <value>123456789</value>
        </data-value>
    </data-values>
    <due-date>Fri Aug 26 21:40:37 PDT 2005</due-date>
    <item-number>1</item-number>
    <price-per-unit>0.0</price-per-unit>
    <priced>true</priced>
    <quantity>1</quantity>
    <rejected>false</rejected>
    <rejected-date/>
    <rejector>
        <company-address/>
        <email/>
        <fax/>
        <first-name/>
        <home-phone/>
        <last-name/>
        <login-name/>
        <person-id>0</person-id>
        <personal-address/>
        <supervisor-name/>
        <timezone/>
        <work-phone/>
    </rejector>
    <requisition-entry-id>1</requisition-entry-id>
    <revision-number>5</revision-number>
    <service>
        <dictionary>
            <data>
                <data-id>3</data-id>
                <data-type>Person</data-type>
                <is-dictionary-external>false</is-dictionary-external>
                <max-length>100</max-length>
                <name>Requester</name>
            </data>
        </dictionary>
    </service>

```

```

</data>
<dictionary-id>1</dictionary-id>
<is-external>>false</is-external>
<is-shared>>false</is-shared>
<name>Monkey Service (private)</name>
</dictionary>
<dictionary>
<data>
<data-id>1</data-id>
<data-type>Text</data-type>
<is-dictionary-external>>false</is-dictionary-external>
<max-length>50</max-length>
<name>TicketID</name>
</data>
<data>
<data-id>2</data-id>
<data-type>Text</data-type>
<is-dictionary-external>>false</is-dictionary-external>
<max-length>50</max-length>
<name>AssetNumber</name>
</data>
<dictionary-id>2</dictionary-id>
<is-external>>false</is-external>
<is-shared>>true</is-shared>
<name>RemedyStuff</name>
</dictionary>
<estimated-cost>0.0</estimated-cost>
<form>
<fields>
<field>
<field-id>2</field-id>
<advanced-prompt/>
<data>
<data-id>2</data-id>
<data-type>Text</data-type>
<is-dictionary-external>>false</is-dictionary-external>
<max-length>50</max-length>
<name>AssetNumber</name>
</data>
<input-type>text</input-type>
<label>AssetNumber</label>
<mandatory>>false</mandatory>
<max-length>50</max-length>
<max-value>0.0</max-value>
<min-value>0.0</min-value>
<multi-select>>false</multi-select>
<options>
<available-keys/>
<available-labels/>
<current-values/>
<multivalued>>false</multivalued>
</options>
<permission>4</permission>
<validated>>true</validated>
</field>
<field>
<advanced-prompt/>
<data>
<data-id>1</data-id>
<data-type>Text</data-type>
<is-dictionary-external>>false</is-dictionary-external>
<max-length>50</max-length>
<name>TicketID</name>
</data>

```

```

    <field-id>1</field-id>
    <input-type>text</input-type>
    <label>TicketID</label>
    <mandatory>>false</mandatory>
    <max-length>50</max-length>
    <max-value>0.0</max-value>
    <min-value>0.0</min-value>
    <multi-select>>false</multi-select>
    <options>
      <available-keys/>
      <available-labels/>
      <current-values/>
      <multivalued>>false</multivalued>
    </options>
    <permission>4</permission>
    <validated>>true</validated>
  </field>
</fields>
<field>
  <advanced-prompt>Give the name!</advanced-prompt>
  <data>
    <data-id>3</data-id>
    <data-type>Person</data-type>
    <is-dictionary-external>>false</is-dictionary-external>
    <max-length>100</max-length>
    <name>Requester</name>
  </data>
  <field-id>3</field-id>
  <input-type>text</input-type>
  <label>Requester Name</label>
  <mandatory>>false</mandatory>
  <max-length>100</max-length>
  <max-value>0.0</max-value>
  <min-value>0.0</min-value>
  <multi-select>>false</multi-select>
  <options>
    <available-keys/>
    <available-labels/>
    <current-values>
      <string>John McGarzafi</string>
    </current-values>
    <multivalued>>false</multivalued>
  </options>
  <permission>4</permission>
  <validated>>true</validated>
</field>
</fields>
<user-id>0</user-id>
</form>
<name>Monkey Service</name>
<parameters>
  <default-duration>0.0</default-duration>
  <priority>2</priority>
  <start-date/>
  <start-mode>0</start-mode>
</parameters>
<pricing-schema>0</pricing-schema>
<quantity>1</quantity>
<service-id>1</service-id>
<version>5</version>
</service>
<start-after/>
<start-date>Wed Aug 24 21:40:50 PDT 2005</start-date>
<start-mode>0</start-mode>
<status>1</status>

```

```

</requisition-entry>
<requisition-id>1</requisition-id>
<started-on>Wed Aug 24 21:40:32 PDT 2005</started-on>
<status>1</status>
<requisition-step>
  <completed-on/>
  <due-on>Fri Aug 26 21:40:37 PDT 2005</due-on>
  <estimated-on>Fri Aug 26 21:40:37 PDT 2005</estimated-on>
  <name>Delivery project for Monkey Service</name>
  <status>2</status>
</requisition-step>
</requisition>
<agent-parameter multi-valued="false">
  <name>Ticket</name>
  <value>None yet</value>
</agent-parameter>
<agent-parameter multi-valued="false">
  <name>Asset</name>
  <value>123456789</value>
</agent-parameter>
</task-started>
</message>

```

take-action (着信)

```

<?xml version="1.0" encoding="UTF-8"?>
<message channel-id="18071221:1124919814742:-32752">
  <take-action action="done"/>
</message>

```

send-parameters (着信)

```

<?xml version="1.0" encoding="UTF-8"?>
<message channel-id="18071221:1116468068789:-32360">
  <send-parameters>
    <agent-parameter>
      <name>Param1</name>
      <value>cat</value>
    </agent-parameter>
    <agent-parameter>
      <name>Param2</name>
      <value>catlitter</value>
    </agent-parameter>
  </send-parameters>
</message>

```

add-comments (着信)

```

<?xml version="1.0" encoding="UTF-8"?>
<message channel-id="32580443:1116276793649:-32629">
  <add-comments>
    <comment>Test Comment</comment>
  </add-comments>
</message>

```