

CBiO Provisioning Customer Adaptation Guide

Ericsson Dynamic Activation 1

USER GUIDE

Copyright

© Ericsson AB 2017. All rights reserved. No part of this document may be reproduced in any form without the written permission of the copyright owner.

Disclaimer

The contents of this document are subject to revision without notice due to continued progress in methodology, design and manufacturing. Ericsson shall have no liability for any error or damage of any kind resulting from the use of this document.

Trademark List

All trademarks mentioned herein are the property of their respective owners. These are shown in the document Trademark Information.



Contents

1	Introduction	1
1.1	Purpose and Scope	1
1.2	Target Groups	1
1.3	Typographic Conventions	1
1.4	Prerequisites	1
2	CBiO Solution Integration Overview	3
2.1	Generic EDIFACT Service Model	4
2.1.1	Northbound	4
2.1.2	Southbound	10
2.2	Action Service Model	11
2.2.1	Northbound	12
2.2.2	Southbound	14
	Reference List	17





1 Introduction

This document describes the processes of developing Business Support and Control System (CBiO) Provisioning with Designer Studio.

1.1 Purpose and Scope

This document is intended for the CA developers to develop CBiO solution integration.

For more information about developing CA with designer studio, see *User Guide for Designer Studio*, Reference [2].

For information about the CBiO provisioning solution, see *Solution Description Charging and CBiO*, Reference [3].

1.2 Target Groups

The target groups for this document are as follows:

- CA developers
- Solution architects

1.3 Typographic Conventions

Typographic conventions are described in the *Library Overview*, Reference [1].

1.4 Prerequisites

The readers of this document must meet the following prerequisites:

- Knowledge of Dynamic Activation and Design Studio
- Knowledge of XML, WSDL.





2 CBiO Solution Integration Overview

The end-to-end CBiO solution integration requires two layers of service models, which are developed by the Designer Studio. Figure 1 shows the service model component in the overall CBiO provisioning solution architecture.

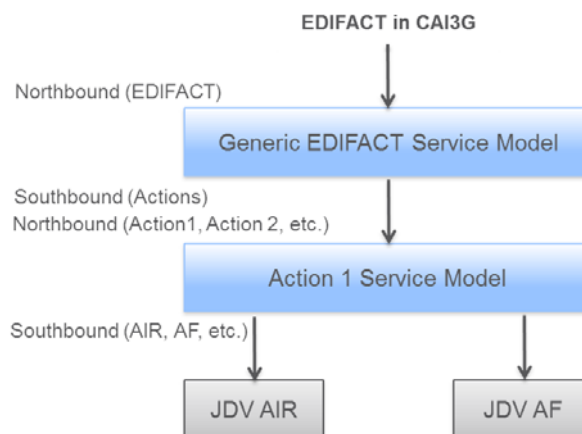


Figure 1 CBiO Provisioning Solution Architecture - Service Model

- **EDIFACT in CAI3G** - It is the Dynamic Activation internal request generated by the Electronic Data Interchange for Administration, Commerce, and Transport (EDIFACT) NBIA from the raw EDIFACT message received from Business Support and Control System (BSCS). EDIFACT in CAI3G is stored in Asynchronous Request handler queue and executed in asynchronous process. CBiO Provisioning only uses a small part of the raw data from the EDIFACT in CAI3G message. It is important to understand the parameter mapping rules in service model design. Because EDIFACT in CAI3G message carries all information in raw EDIFACT message, even the information is redundant or useless, to ensure the data integrity.
- **Generic EDIFACT Service Model (Layer 1 Service Model)** - It is the first layer of the service model of the CBiO business logic. It dispatches the requests into various Layer 2 service models based on the action number. The northbound of this service model must be customized according to the customer CBiO solution. The Southbound of this service model is the northbound of the Layer 2 service models for each EDIFACT action.
- **Action 1 Service Model (Layer 2 service model)** - It represents the CBiO business logic dedicated to handle specific EDIFACT action. The Northbound of this service model must be customized according to the customer CBiO solution. The Southbound of this service model can be the WSDL and schema of the NE provisioning, or other business logic (layer N [N>2] service models, subscriber view JDVs).



2.1 Generic EDIFACT Service Model

This sections give the explanation of the elements in EDIFACT Service Model templates and the detailed implementation guide on how to extend the EDIFACT Service Model template base on the customer requirement. For detailed information about the Designer Studio, refer to the *User Guide for Designer Studio*, Reference [2].

To get the example EDIFACT Service Model templates, follow below instruction:

1. Save the [CBiO_Service_Model_examples.zip](#) file, to a temporary location.
2. Unpack the zip file.

When unzipping, there are three folders: Generic EDIFACT Service Model, Action1 Service Model, and Action8 Service Model.

3. Get the templates from the Generic EDIFACT Service Model folder and add them in a suitable location.

2.1.1 Northbound

The EDIFACT NBIA has a set of rules to parse the raw EDIFACT message into a structured CAI3G message. In the EDIFACT Service Model template, this CAI3G message is called the **EDIFACT in CAI3G**.

The schema template for Generic EDIFACT Service Model includes the following files:

- EDIFACT.wsdl
- EDIFACT.xsd - Describes the detailed definition of the template format.
- cai3g1.2_provisioning.xsd (Generic CAI3G included in EDIFACT.xsd)

It is recommended to follow the structure in template to customize the Northbound Interface for EDIFACT Service Model. For detailed information about the EDIFACT interface, refer to *Generic EDIFACT Interface Specification*, Reference [4].

The following sections give the detailed description of the `createSubscripition` template in `EDIFAT.xsd` and the guide on how to customize `createSubscripition` schema.

2.1.1.1 Create Subscription

Figure 2 and Figure 3 show the parameters of the `createSubscription` template.

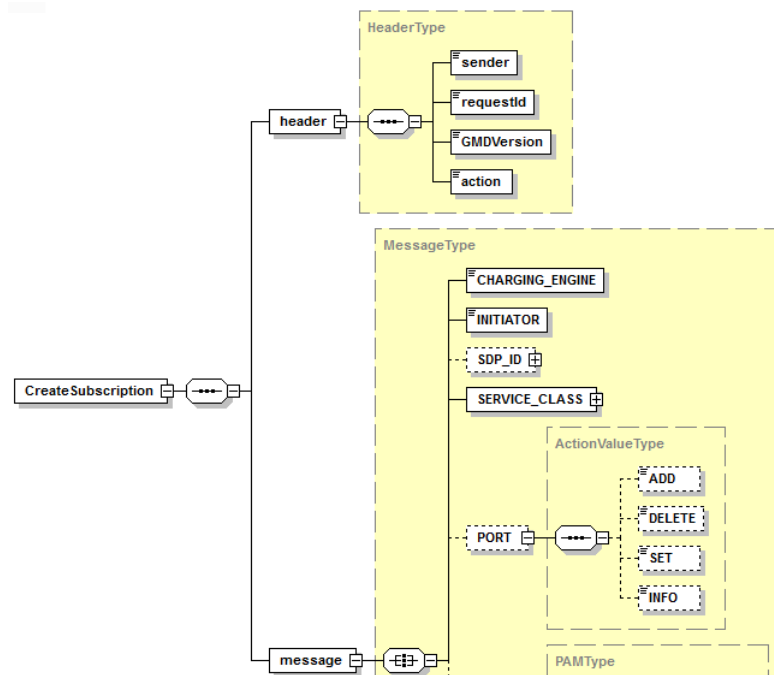


Figure 2 Create Subscription (Part 1)

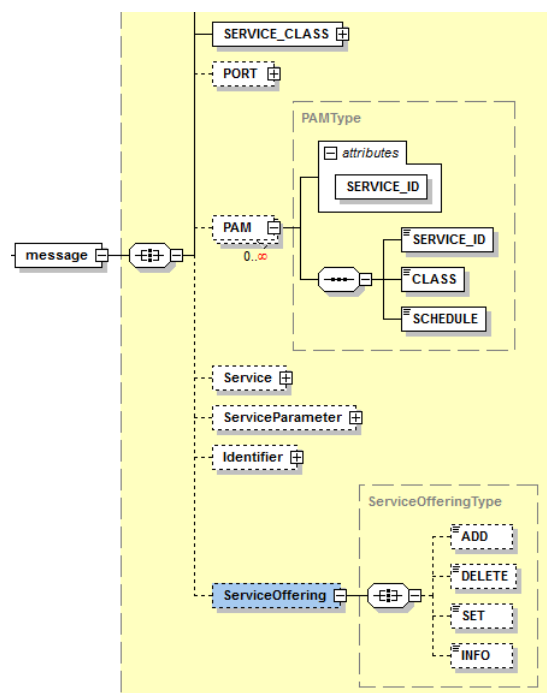


Figure 3 Create Subscription (Part 2)

2.1.1.1.1 Header

The Header element contains four sub-parameters: sender, requestId, GMDVersion, action.



No default value needs to be changed in Header.

The parameter `action` is used as the **Condition** of the target tasks in the service model. For how to configure the condition of the target task, refer to Section 2.1.2.1 on page 10.

2.1.1.1.2 Message

The `Message` element contains all the services and service parameters information in the raw EDIFACT message.

In `Message` element, some of the sub-parameters are the fixed parameters while other elements need to be extended to fit the customer requirement.

2.1.1.1.3 CHARGING_ENGINE

`CHARGING_ENGINE` is a fixed parameter. This parameter is translated from the `CHARGING_ENGINE` in EDIFACT message to indicate where the subscriber is managed or not. The possible value can be `BSCS` or `CS`.

It is commonly used by the task condition of Layer 2 service models. Therefore, the value needs to pass through the layer 1 service models.

2.1.1.1.4 INITIATOR

`INITIATOR` is a fixed parameter. This parameter is translated from the `INITIATOR` in EDIFACT message to indicate where the request is initiated. The value can be `CC` or `BATCH`.

It is commonly used by the task condition of Layer 2 service models. Therefore, the value needs to pass through the layer 1 service models.

2.1.1.1.5 Contract Parameters

Figure 4 shows the contract parameters in the Create subscription template.

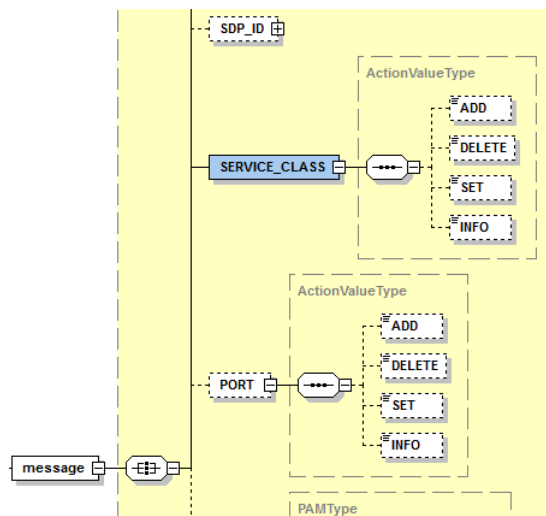


Figure 4 Contract Parameters

Contract Parameters represent the message belong to a subscription instead of a service. For example, `SDP_ID`, `SERVICE_CLASS` and `PORT` in the `createSubscription` template. The Contract Parameters translated from the EDIFACT are stored under the `message` element. The contract parameters need to be modified according to the custom CBiO implementation.

It is commonly used by Layer 2 service models to provisioning certain network elements. Therefore, the value needs to pass through the layer 1 service models.

2.1.1.1.6 Periodic Account Management

PAM is a fixed element. Periodic Account Management (PAM) is the sub-MO translated from the contract parameter `set` in EDIFACT message. Each PAM has three sub-parameters `service id`, `class id`, and `schedule id`.

It is used by Layer 2 service models to provisioning CS or AIR. Therefore, the value needs to pass through the layer 1 service models.

2.1.1.1.7 Service

Figure 5 shows the `Service` element in the Create subscription template.

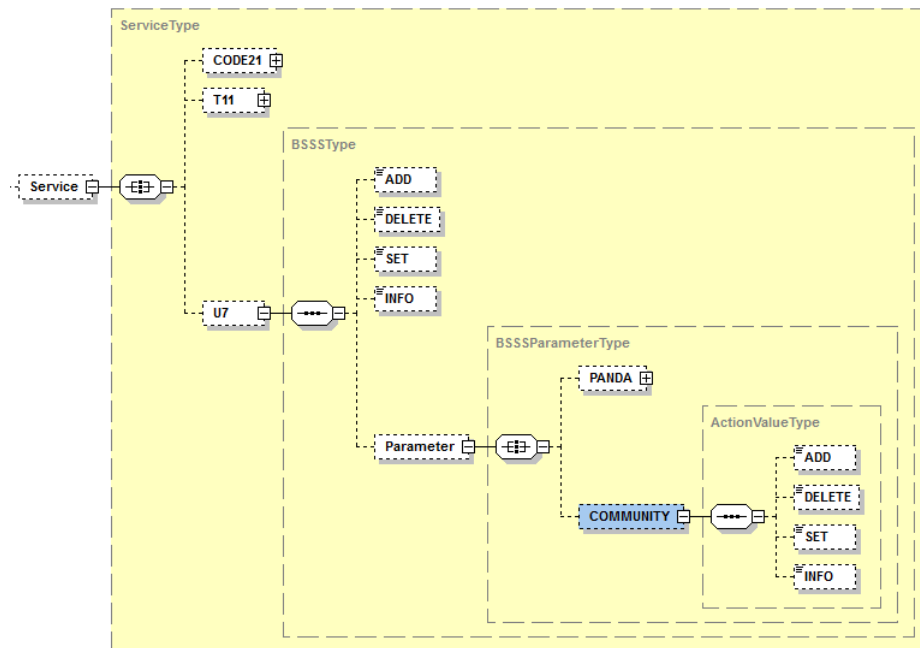


Figure 5 Service

The parameters under `Services` elements are translated from the raw EDIFACT message. For example, T11, U7 and CODE21 in the template. When translating the service name that only has digits, add `CODE` before the digits. Because XML does not allow using digits as the element name.

The `Service` parameters need to be modified according to the custom CBiO implementation by modifying and adding new services and their parameters.

It is commonly used by Layer 2 service models to provisioning certain network elements. Therefore, the value needs to pass through the layer 1 service models.

2.1.1.1.8

Service Parameter

Figure 6 shows the `ServiceParameter` element in the Create subscription template.

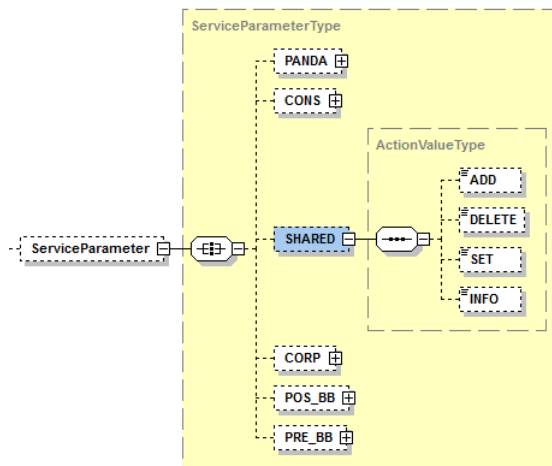


Figure 6 Service Parameter

The parameters enclosed in this `ServiceParameter` element are different from the parameters enclosed in the `Service` element. It contains the parameter translated from the parameters under a service (XSV) but without a service name (BS and SS fields are empty) in the raw EDIFACT message.

The `ServiceParameter` parameters need to be modified according to the custom CBIo implementation.

It is commonly used by Layer 2 service models to provisioning certain network elements. Therefore, the value needs to pass through the layer 1 service models.

2.1.1.1.9 Identifier

Figure 7 shows the `Identifier` element in the Create subscription template.

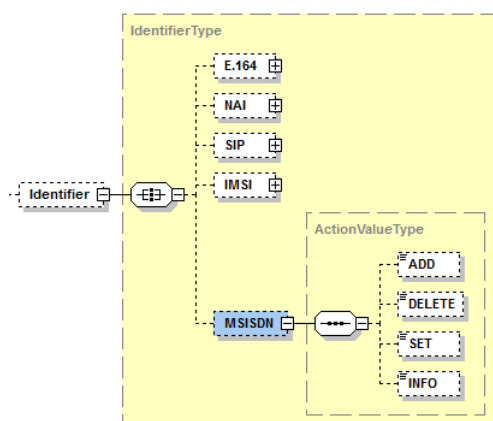


Figure 7 Identifier

`Identifier` is a fixed element. The identifiers are translated from the `RESOURCE` parameters in the raw EDIFACT message.

It is commonly used by the Layer 2 service models to provisioning certain NEs. Therefore, the value needs to pass through the layer 1 service models.

2.1.1.1.10 ServiceOffering

`ServiceOffering` is a fixed element. It contains all the `SERVICE_OFFERING` and `SERVICE_OFFERING_SP` values which are collected from the raw EDIFACT message and filled into the value of the `operation` element. The values are separated by “,”.

It is commonly used by Layer 2 service models to provisioning CS or AIR. Therefore, the value needs to pass through the layer 1 service models.

2.1.2 Southbound

The Southbound of the Generic EDIFACT Service Model is the WSDL and schemas of each action supported. They are also acting as the northbound of the Action Service Models. For example, the **Action 1 Service Model** described in Section 2.2 on page 11.

2.1.2.1 Service Model Design

The Generic EDIFACT Service Model is used to dispatch information in the **EDIFACT in CAI3G** to the corresponding Action Service Models.

The **Conditions** for each task is configured according to the value of the `action` element in `EDIFACT.xsd`.

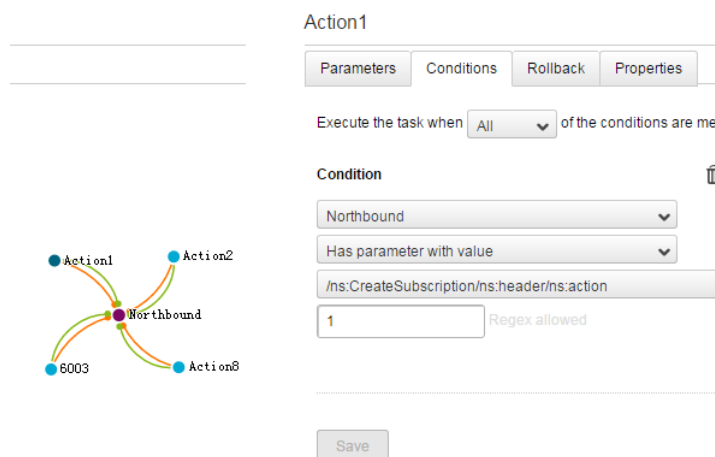


Figure 8 Designer Studio - Conditions

Both of the parameters used to define the execution condition and provision the NE needs to be connected.



For example, `initiatorId` and `chargingEngine` are connected to the `INITIATOR` and `CHARGING_ENGINE` of the northbound as shown below. They are commonly used for the condition of task to decide whether the Charging System shall be provisioned.

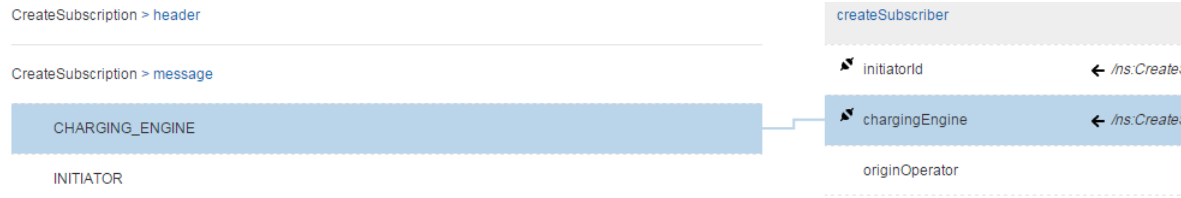


Figure 9

IMSI and MSISDN are often used as the key attributes in provisioning various NEs. They are connected to the value of respective operation in the Identifier element of the northbound as shown below.

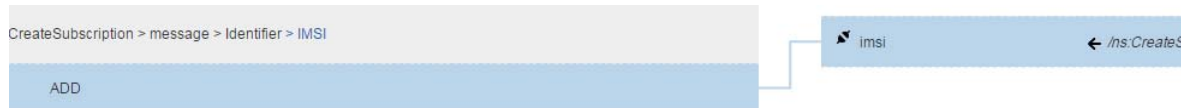


Figure 10

IMSI can also be connected to the `PORT` value in `ADD` operation for Action 1. Some parameters can appear in multiple places in the northbound because there is redundant information in the raw EDIFACT message. Solution Integrator needs to choose the most accurate and convenient way of parameter connection.



Figure 11

2.2 Action Service Model

Action Service Models replaced the Subscriber Views to manage the provisioning flow towards NEs. Action 1 Service Model is an example for installing a new subscription in CBiO Solution.

To get the example of Action1 Service Model templates, follow below instruction:

1. Save the [CBiO Service Model examples.zip](#) file, to a temporary location.
2. Unpack the zip file.

When unzipping, there are three folders: Generic EDIFACT Service Model, Action1 Service Model, and Action8 Service Model.

3. Get the templates from the Action1 Service Model folder and add them in a suitable location.

2.2.1 Northbound

The northbound of Action Service Models is designed to meet the provisioning requirements of NEs it managed.

The schema template for Action1 Service Model in the Northbound Interface includes the following files:

- Action1.wsdl
- Action1.xsd
- cai3g1.2_provisioning.xsd (generic CAI3G included in EDIFACT.xsd)

The following schema is the part of the Action1.xsd Solution integrator. It needs to be updated with customer-specific parameters.

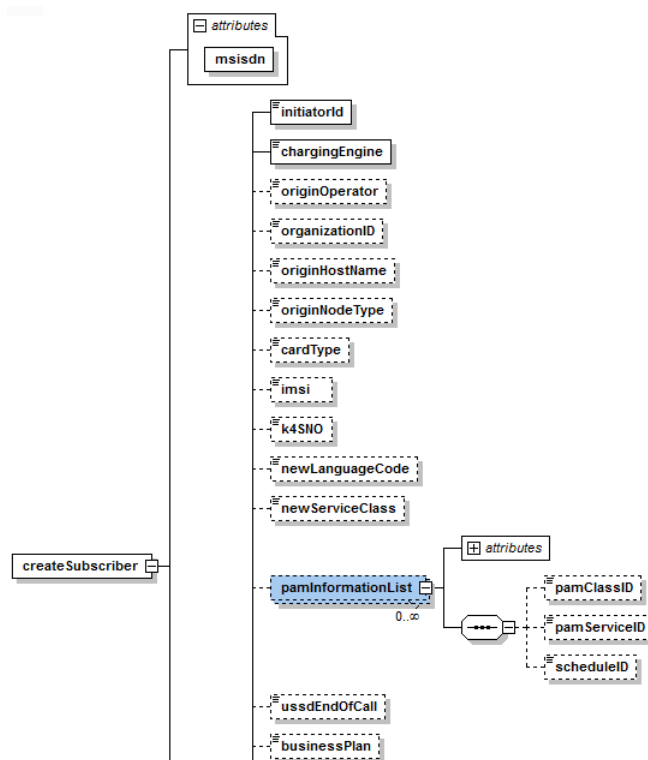


Figure 12 Action1.xsd (Part 1)

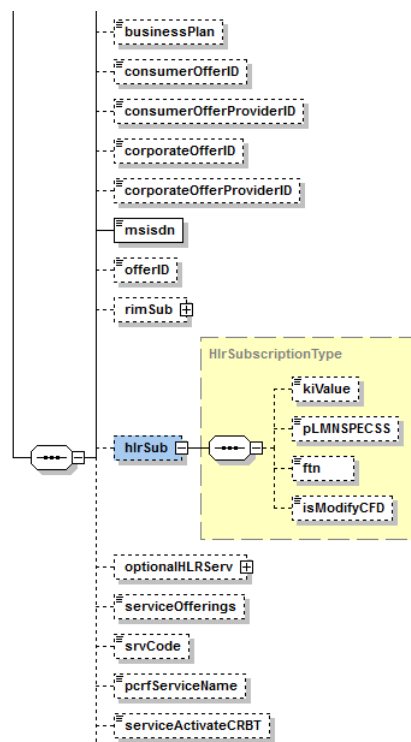


Figure 13 Action1.xsd (Part 2)



Figure 14 Action1.xsd (Part 3)

2.2.2 Southbound

The Southbound of Action Service Models are the NE provisioning interfaces in CAI3G supported by Dynamic Activation, either in standard or by Customer Adaptation. In the Active1 Service Model template, only the provisioning of AIR and AF of Ericsson Charging System have been implemented.

2.2.2.1 Service Model Design

The following figure shows the template of Action1 service model. AF is provisioned first to create MSISDN and other identifiers, and then AIR is provisioned to update communities. Solution Integrators need to update the service model according to customer requirements.

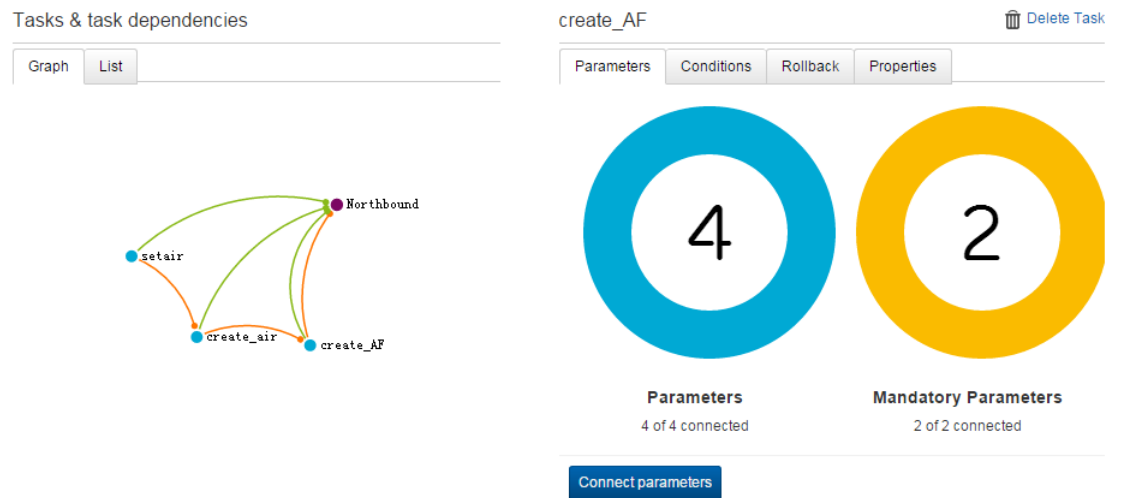


Figure 15 Designer Studio - Action1 Service Model

The parameters in the northbound of Action Service Models are connected from the parameters of the **EDIFACT in CAI3G** in the Generic EDIFACT Service Model. They have to be further connected to the southbound NE parameters. The following figure shows an example of connecting parameters within a sub-MO in the Active1 Service Model template.



Figure 16 Designer Studio - Connect Parameters





Reference List

Ericsson Documents

- [1] *Library Overview*, 1/1553-2/CSH 109 554 Uen
- [2] *User Guide for Designer Studio*, 10/1553-2/CRH 109 1438 Uen
- [3] *Solution Description Charging and CBiO*, 1/221 02-2/CRH 109 1438 Uen
- [4] *Generic EDIFACT Interface Specification*, 3/155 19-2/CRH 109 1438 Uen