

Function Specification Layered EIR

Ericsson Dynamic Activation 1

FUNCTION SPECIFICATION

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 Group	1
1.3	Typographic Conventions	1
2	Layered EIR Provisioning Solution	2
2.1	Overview	2
2.2	Data Model EIR-FE	3
2.3	Atomicity and Integrity Handling	5
2.4	EIR Provisioning	5
2.4.1	Equipment / Equipments	6
2.4.2	Search Order	6
2.4.3	Equipment Status	7
2.4.4	Mobile Station	8
2.4.5	Cloned Handset User	9
2.4.6	Fixed SIM	9
2.5	EIR Massive Searches	9
2.5.1	Equipment	10
2.5.2	Search Orders	10
2.5.3	Cloned Handset Users	11
2.5.4	Fixed SIM	11
	Reference List	13





1 Introduction

This section is an introduction to this document. It contains information about the prerequisites, purpose, scope, and target group for the document. This section also contains explanations of typographic conventions used in this document.

1.1 Purpose and Scope

This document gives a brief introduction to the Layered Equipment Identity Register (EIR) provisioning solution, provided by Ericsson™ Dynamic Activation (EDA).

1.2 Target Group

The target group for this document is as follows:

- Application Administrator
- Marketing
- Network Administrator
- System Administrator
- Network Supervision Administrator
- Application Designer

For more information regarding the different target groups, see *Library Overview*, Reference [3].

1.3 Typographic Conventions

Typographic conventions are described in *Library Overview*, Reference [3].

2 Layered EIR Provisioning Solution

Data Layered Architecture (DLA) is an Ericsson architecture that provides a layered structure for network elements which allows separation of traffic logic and data storage into different nodes.

2.1 Overview

An overview of the EIR provisioning solution and including nodes is shown in Figure 1.

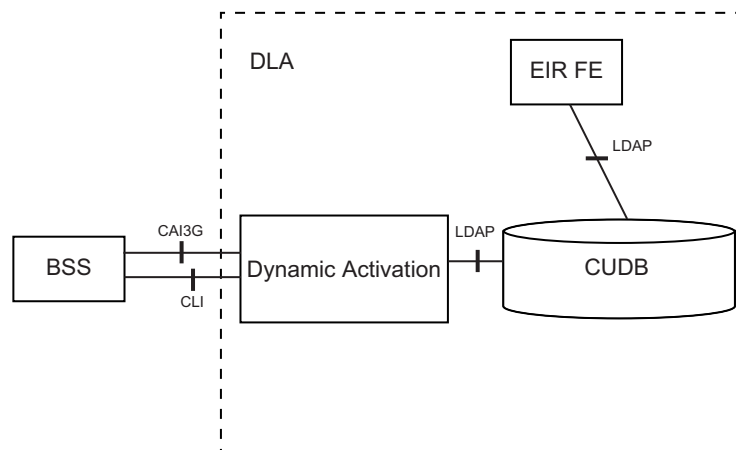


Figure 1 DLA EIR Overview

BSS Business Support System (BSS) initiates the provisioning request towards Dynamic Activation.

Dynamic Activation/Provisioning Gateway

A provisioning system that provides a single provisioning interface towards the BSS, by hiding the complexities of provisioning underlying network.

CUDB The Back-End (BE) database, offered by the Ericsson's realization of DLA, which decouples the user data storage from the application logic in the Front-Ends (FEs). All data associated with layered EIR is stored in Centralized User Database (CUDB).



EIR-FE

EIR is an equipment register node that supports the network operators to control illegal use of mobile equipments in their network. When a mobile phone register to the network, the following happens:

- The Mobile Switching Center (MSC), serving the location of the phone, sends a `CheckIMEI` request to the EIR-FE.
- The `CheckIMEI` request includes an IMEI, which is the unique international mobile station equipment identity, uniquely identifying the equipment part of the mobile phone.
- EIR generates a status for IMEI, based on the different lists that can be defined for an IMEI, which the MSC uses to accept or blacklist the equipment.

EIR core functionality is `CheckIMEI` processing.

2.2 Data Model EIR-FE

The following figure shows the EIR-FE provisioning data model in CUDB.

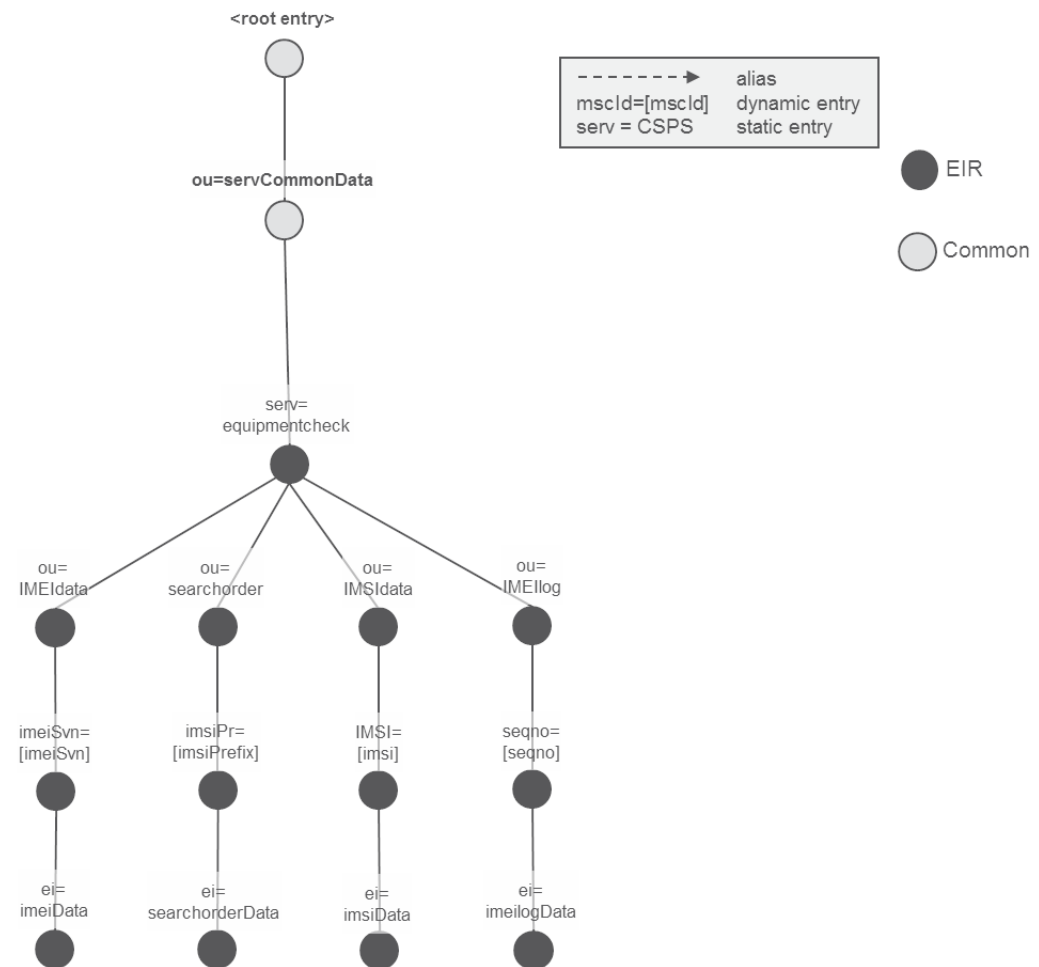


Figure 2 Data Model EIR-FE

Dynamic Activation is responsible for the provisioning functions:

- Single point of provisioning

Map the CAI3G order to the LDAP objects in the data model before provisioning.

- Syntax, schematics check and validation of provisioned data

Check that the provisioned data is of correct type and add default values, if needed, to attributes that are required (mandatory) in CUDB but optional in CAI3G. Performs the nomination check for valid cases.

- Keep data consistency in CUDB. For more information, see Section 2.3 on page 4.



2.3 Atomicity and Integrity Handling

Atomicity means ensuring that any operations performed on the system are either all completed successfully or all reversed successfully to keep the data consistency.

One CAI3G Customer Service Orders (CSO) can imply several Lightweight Directory Access Protocol (LDAP) orders towards the CUDB. Dynamic Activation will provide atomicity in EIR provisioning as below:

- Parses and validates the whole CSO before any LDAP order is sent towards the CUDB to minimize the LDAP errors received from the CUDB.
- Retry the LDAP order when some LDAP errors are returned from CUDB, for example Function Busy. The number of retries is configurable. For more information about retry setting, refer to *User Guide for Resource Activation*, Reference [6].

During the provisioning process, an error can occur that leads to partial execution of a provisioning operation. These errors are related to writing of nomination log records in CUDB. These outstanding LDAP commands are logged in the `eir_partiallysucceededoperations.log` file in Dynamic Activation. Manual repair actions must be performed directly towards the CUDB.

For more information about repair procedure, refer to Section 2.4.1 on page 5 and *CUDB Subscription Repair and Remove Procedures*, Reference [4].

Note: Simultaneous Create, Set and Delete of the same equipment data can result in inconsistent data in the CUDB. Reserve sufficient time, with consideration to retry behavior, between the different operations.

2.4 EIR Provisioning

For provisioning of Layered EIR data, CAI3G is the offered interface. Through the CAI3G interface it is possible to perform the Create, Set, Get, and Delete operations for the following CSOs:

- Create/Set/Get/Delete Equipment
- Create/Delete Equipments
- Create/Set/Get/Delete SearchOrder
- Get EquipmentStatus
- Create/Get/Delete MobileStation
- Create/Set/Get/Delete ClonedHandsetUser
- Create/Get/Delete FixedSim

For information about CAI3G layered EIR provisioning, refer to *Layered EIR Provisioning over CAI3G*, Reference [1].

2.4.1 Equipment / Equipments

Both the `Equipment` and the `Equipments` MOs are handling the provisioning of `IMEIs` (the identity of the equipment part of the mobile phone) in different lists. They write to the `IMEIdata` and, depending on the use case, also to the `IMEIlogdata` objects in the CUDB.

The `IMEIlogdata` object is written when `Nomination` is activated, meaning the operator has a connection towards the centralized `IMEI` database. EIR FE synchronizes the `IMEIlogdata` towards the centralized `IMEI` database regularly. This centralized `IMEI` database includes the status of all equipment from all operators in the world that has a connection towards it. This means that stolen equipment cannot be used in any of these operator networks. When `Nomination` is active and the equipment is defined or removed from a predefined `Nomination-list`, the log object must be written to keep this centralized `IMEI` database up to date.

If it, for some reason, was not possible to complete the operation towards the `IMEIlogdata` object, the outstanding CUDB operations are written in the `eir_partiallysucceededoperations.log` file in Dynamic Activation. This log file can be used for manual update of the CUDB with correct values before the centralized `IMEI` database is synchronized. For information on how to perform these manual updates, see *CUDB Subscription Repair and Remove Procedures*, Reference [4].

The differences between the `Equipment` and the `Equipments` MOs are that `Equipment` operates on one single `IMEI` object, while `Equipments` can operate on 1-10 `IMEI` objects at the same time, defining them the same way.

The `create` operation defines an `IMEI` for the first time. The `Get` operation returns the content of the requested `Equipment` (`IMEI`) object. The `Set` operation changes the content (add to list or remove from list) of an already defined object. The `Delete` operation removes the complete object from the CUDB.

2.4.2 Search Order

The `SearchOrder` MO is handling the provisioning of search orders. `IMSIprefix` are associated with different list that is to be searched to find out the equipment status. It writes to the `searchorderData` object in the CUDB.

The `create` operation defines a search order, with up to five lists defined for the first time. The `Set` operation changes the content (add or remove a list or the response value of a list) of one already defined object. The `Delete` operation removes the complete object from the CUDB.

The `Get` operation uses below process to find out the matching search order in the database.



An incoming IMSIprefix can match prefixes associated with more than one search order. In this case, the search order used is IMSIprefix with the longest matching digits of the IMSI. As an example, the following search orders and IMSIprefixes are available:

- 1234 with list 0 (white), list 1 (black) and list 6 (grey) and unknown response White
- 1234568 with list 5 (white), list 1 (black) and list 2 (grey) and unknown response Black
- 1234567 with list 4 (white), list 1 (black) and list 2 (grey) and unknown response Black
- 12345678 with list 6 (white), list 1 (black) and list 2 (grey) and unknown response Black

The following IMSI arrive:

- IMSI 123456789012345.

IMSI 123456789012345 matches the following search orders:

- search order with IMSIprefix 1234 (first 4 digits of the IMSI match this prefix)
- search order with IMSIprefix 1234567 (first 7 digits of the IMSI match this prefix)
- search order with IMSIprefix 12345678 (first 8 digits of the IMSI match this prefix)

IMSI 123456789012345 does not match the search order with IMSIprefix 1234568, because digit 7 of the IMSI do not match the IMSIprefix n that search order.

The best match is with the IMSIprefix 12345678, because this is longer than the other matching prefixes.

Since the incoming IMSIprefix can be shorter than 15 digits it means it can be shorter than the defined imsiPrefixes in the custom search orders. In this case the search order is IMSI prefix with the largest number.

For example, the search order for IMSIprefix 121 is requested.

There exist search orders with the IMSIprefixes 121, 1213455 and 1213456.

In this case it is matched against the IMSIprefix as follows:

- The supplied IMSIprefix 121 matches all the custom search orders.
- The custom search orders for 121, 1213455 and 1213456 match all three digits of the supplied IMSIprefix.
- The custom search order 1213456 is the largest number, so selected as the best match.

Note: Dynamic Activation expects no more than 100 search order records to be present. For the `Get` operation a security limit of 1000 records is defined in Dynamic Activation. If security limit reached and `Get` search order is attempted, an error response is delivered.

There is also a default search order that must exist in the CUDB when EIR FE is connected. This default search order is not allowed to be created or deleted through Dynamic Activation. However it is possible to change the content of this record with the `Set` operation and present the object with the `Get` operation.

2.4.3

Equipment Status

This `EquipmentStatus` operation fetches values from both the `IMEIdata` and the `SearchOrderdata` objects, combine the answers and present the status of the equipment. The status can be Black, Grey, White, or Unknown.

If the `Get` operation includes an IMSI, then a matching search order is searched for and related lists are searched in the `IMEI` object. If the `IMEI` is set in a list pointed to from the search order, the response of that search order list is returned. If no list is set, the value in the unknown response of that search order is returned.



If the `Get` operation do not include an `IMSI`, the default search order is used and the same procedure is used as described above.

When this `Get` operation finds the matching search order, the following process is used:

An incoming `IMSIprefix` can match prefixes associated with more than one search order. In this case, the search order used is `IMSIprefix` with the longest matching digits of the `IMSI`. As an example, the following search orders and `IMSIprefixes` are available:

- 1234 with list 0 (white), list 1 (black) and list 6 (grey) and unknown response White
- 1234568 with list 5 (white), list 1 (black) and list 2 (grey) and unknown response Black
- 1234567 with list 4 (white), list 1 (black) and list 2 (grey) and unknown response Black
- 12345678 with list 6 (white), list 1 (black) and list 2 (grey) and unknown response Black

The `IMEI` 11111122555555 with default `SVN` is in lists 1 and 6. The following `IMEI` and `IMSI` combination arrive:

- `IMEI` 11111122555555 with `IMSI` 123456789012345.

`IMSI` 123456789012345 matches the following search orders:

- search order with `IMSIprefix` 1234 (first 4 digits of the `IMSI` match this prefix)
- search order with `IMSIprefix` 1234567 (first 7 digits of the `IMSI` match this prefix)
- search order with `IMSIprefix` 12345678 (first 8 digits of the `IMSI` match this prefix)

`IMSI` 123456789012345 do not match the search order with `IMSIprefix` 1234568, because digit 7 of the `IMSI` do not match the `IMSIprefix` in that customized search order.

The best match is with the `IMSIprefix` 12345678, because this is longer than the other matching prefixes.

The search order for that prefix says that list 6 is the first list to search. `IMEI` 11111122555555 is indeed in list 6, so the corresponding response (white) is chosen for the `IMEI/IMSI` combination. Thus the status response for the `IMEI/IMSI` combination is white and the list matched is 6.

An example where the `IMSI` provided in the request is shorter than 15 digits the matching of search orders can as an example be:

`IMSI` provided 123456, `IMSIprefix` available are 123,, 12345, 123456 and 1234567.

- The supplied `IMSI` 123456 matches the custom search orders prefixes 123, 12345 and 123456.
- The custom search order for 123456 is the longest matching prefix and is the one chosen.
- The custom search order 1234567 are not matching, the whole prefix must match the `imsi`.

The full `IMSI` from the customer input is matched with the longest matching `IMSIprefix` in the available search orders.

Note: Since this `Get` operation, also fetches search order data records (where `Dynamic Activation` expects no more than 100 records), a security limit of 1000 records is defined in `Dynamic Activation`. If security limit reached and `Get` equipment status is attempted, an error response is delivered.

2.4.4

Mobile Station

The `MobileStation MO` is handling the provisioning of `lockedIMSI`, the `IMSI` that is related to the `Equipment`. It writes to the `IMEIdata` objects in the `CUDB`.



The `Create` operation defines a `lockedIMSI` to the equipment. If the equipment does not exist, it is created. This means that this equipment can only be used with this IMSI. The `Delete` operation removes the `lockedIMSI` to an equipment object. If the equipment is not connected to a list, then the whole object is removed from the CUDb. The `Get` operation returns the `lockedIMSI` related to requested equipment (IMEI).

2.4.5 Cloned Handset User

The `CloedHandsetUser` MO is handling the provisioning of `ClonedHandsetUsers` (Equipment (IMEIs) cloned to a specific IMSI). It writes to `IMSIdata` objects in the CUDb.

The `Create` operation defines equipment (IMEIs) cloned to an IMSI for the first time. This means that this IMSI can only be used in these (1-5) equipment (IMEIs). The `Set` operation changes the content (add or remove equipment (`clonedIMEIs`) of one already defined object.

The `Delete` operation removes the complete object from the CUDb if no `lockedImeiSvn` exists for the IMSI. If a `lockedImeiSvn` entry exists for the IMSI, then only the `clonedImeiSvn` data is removed.

The `Get` operation returns the cloned equipment (IMEIs) related to requested IMSI.

2.4.6 Fixed SIM

The `FixedSim` MO handles the provisioning of `lockedImeiSvn`, the IMEI that is locked to an IMSI. It writes to the IMSI data objects in the CUDb.

The `Create` operation defines a `lockedImeiSvn` to the IMSI. If the IMSI data does not exist, it is created. This means that the IMSI can only be used with this equipment.

The `Delete` operation removes the `lockedImeiSvn` from the IMSI. This means that the IMSI can be used in other equipment. If `clonedImeiSvn` data exists for this IMSI, then only the `lockedImeiSvn` data is removed from the `IMSIdata` object. If no `clonedImeiSvn` exists for this IMSI, then the whole `IMSIdata` object is removed from the CUDb.

The `Get` operation returns the `lockedImeiSvn` related to the requested IMSI.

2.5 EIR Massive Searches

The massive operations are search operations of EIR data in the CUDb. Such operations are supported through the provisioning Command-Line Interface (CLI).

A conditional search example is shown in Figure 3.

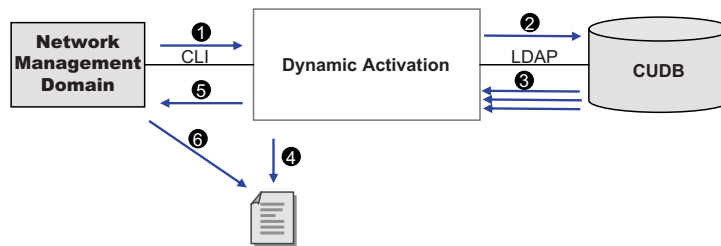


Figure 3 Conditional Search

1. The conditional search task is received.
2. The LDAP search command is sent to the CUDB.
3. The CUDB sends back all found data that matches the search condition.
4. The results are stored in an Extensible Markup Language (XML) file.
5. The notification is sent to the client when the task is completed.
6. The client fetches the file by using the SSH FTP (SFTP).

Following massive search operations are supported for EIR data:

- EEMSEQP
- EEMSSOP
- EEMSCLP
- EEMSFSP

For information, refer to the following documents:

- *Layered EIR Massive Operations over CLI*, Reference [2]
- *Generic CLI Interface Specification*, Reference [5]

2.5.1 Equipment

```
EEMSEQP:IMEIALL[,LIST=list];
```

This command prints all equipments or all the equipments active in a specified list. The operation fetches data from the `IMEIdata` object in the CUDB.

The result, which can be fetched by the client, is stored in an XML file in Dynamic Activation.

2.5.2 Search Orders

```
EEMSSOP;
```



This command prints all search orders. The operation fetches data from the `searchorderData` object in the CUDB.

The result, which can be fetched by the client, is stored in an XML file in Dynamic Activation.

2.5.3 Cloned Handset Users

`EEMSCLP;`

This command prints all cloned handset users. The operation fetches data from the `IMSIdata` object in the CUDB. It returns 1-5 equipment allowed for each specific IMSI.

The result, which can be fetched by the client, is stored in an XML file in Dynamic Activation.

2.5.4 Fixed SIM

`EEMSFSP;`

This command prints all Fixed SIM entries. The operation fetches data from the `IMSIdata` object in the CUDB. It returns the equipment locked to each specific IMSI.

The result, which can be fetched by the client, is stored in an XML file in Dynamic Activation.





Reference List

Ericsson Documents

- [1] *Layered EIR Provisioning over CAI3G*, 18/155 19-CSH 109 628 Uen
- [2] *Layered EIR Massive Operations over CLI*, 17/155 19-CSH 109 628 Uen
- [3] *Library Overview*, 18/1553-CSH 109 628 Uen
- [4] *CUDB Subscription Repair and Remove Procedures*, 4/1553-CSH 109 628 Uen
- [5] *Generic CLI Interface Specification*, 15/155 19-CSH 109 628 Uen
- [6] *User Guide for Resource Activation*, 1/1553-CSH 109 628 Uen