



## INTERFACE GUIDE

# Product 10.5

## File-Based Interface Guide

Copyright © 2005 All rights reserved.

Publication date: April 2008

Document Revision: 01

---

# Summary of Revisions

## **Product File-Based Interface Guide**

825-0671-010

The following changes were made to this document since the last version:

<b>11//05</b>	Initial release of this document.
---------------	-----------------------------------





# Contents

- + **File-Based Interface Guide** .....7
- Overview .....8
- Configuring the File-Based Interface Process .....8
- Starting the File-Based Interface Process .....8
- Formatting Charging Request Files .....9
- Copying the Charging Requests .....10
- Processing File-Based Interface Files .....11
- Viewing Response Files .....12
- Investigating Results .....13
- Glossary of Terms .....15



# File-Based Interface Guide

This document contains the following topics:

- + [Overview](#)
- + [Formatting Charging Request Files](#)
- + [Processing File-Based Interface Files](#)
- + [Viewing Response Files](#)
- + [Glossary of Terms](#)

## Overview

This document describes the specifications required to develop files that can be processed by the Product GCE for charging non-voice, prepaid wireless services via a file-based interface. This file-based interface has been developed to handle charging requests coming into the Product system from service providers who do not support the socket-based, Company-proprietary GCAP interface. Before the files can be processed, the service provider must convert the charging requests into a standard GAP format as published by Company. For the format details, see [Formatting Charging Request Files](#).

The files contain charging requests for non-voice services that have already been delivered to prepaid service consumers. Product uses its proprietary `gcpFileIntf` process to:

- + look for and read charging request files after they are copied to a designated subdirectory, in real time.
- + send the messages to the GCE, where they are approved or denied and the prepaid accounts are charged accordingly.
- + make the results available to service providers in response files, which report successful and unsuccessful (denied) charging responses.
- + move the files whose records have been sent for charging to a designated subdirectory for later purging.

## Configuring the File-Based Interface Process

The `gcpFileIntf` process must be enabled on the SCP-D (Service Control Point - Data) before it can be run. Log in as Administrator, open the `process_control_entities.xml` in a Unix editor, and configure the following:

```
<!ENTITY gcpFile.enabled          "TRUE">
```

---

**WARNING** Do not change the value of the `gcpFile.totalNwiNumber` parameter from 1, or the process will not work correctly.

---

## Starting the File-Based Interface Process

The `gcpFileIntf` process must be up and running in order to process the files. To check the status of the process, enter:

```
status_xxxx DATA.GCPFILE.INTF.gcpFileIntf.0
```

If the process is not running, to start the `gcpFileIntf` process, enter:

```
start_xxxx DATA.GCPFILE.INTF.gcpFileIntf.0
```

## Formatting Charging Request Files

In order for Product to process the charging requests using the file-based interface to the GCE, service providers must convert the charging requests into a standard GAP format. The service provider can include as many charging request messages as needed in the same file.

**IMPORTANT!** The files containing the messages must be named using Company's naming convention (for example, **chargingMsgMMDDYYHHMMSS.in**). Although the file names are configurable, each file name must be unique and must have the extension **.in**.

The following rules apply to creating the GAP-formatted message:

- + The **GapMessage number** parameter must be unique for every message in the file. Replace **n** with the number 1 in the first message, and increment each subsequent message's **GapMessage number** parameter by 1.
- + The **RatingInvoke version**, **Operation**, **IsPost**, **ExternalNodeId**, and **NodeServiceId** parameters are mandatory and their values must be entered exactly as shown in [Figure 1-1](#). The other values can be changed as needed.

1. Create the GAP-formatted message as follows, inserting your charging request values where applicable.

```
MSG                               ↵
<GapMessage type="REQ" number="n">↵
<RatingInvoke version="1.0">↵
<Operation>CHARGE</Operation>↵
<IsPost>>true</IsPost>↵
<ExternalNodeId>FLEX</ExternalNodeId>↵
<NodeServiceId>GCPSVC</NodeServiceId>↵
<MdnMsisdn>9495553000</MdnMsisdn>↵
<MonetaryAmount>10</MonetaryAmount>↵
</RatingInvoke>↵
</GapMessage>
```

Figure 1-1 Sample GAP-Formatted Message

2. Count the number of bytes that make up each message. Enter:

```
wc -c <filename>
```

3. Add 36 to the count.
4. Create a header in the GAP formatted message as shown in the sample header below, inserting the byte count + 36. **This header must be 31 characters long, including spaces.** Add a newline (also known as line feed) character to the end of the header as shown below.

```
MSG nn                               ↵
```

5. Create a trailer at the end of the message. Insert a newline character after **</GapMessage>** and enter:

```
END
```

The message with header and trailer must match the pattern shown in [Figure 1-2](#).

```
MSG 86                               ↵
<GapMessage type="REQ" number="1">↵
<RatingInvoke version="1.0">↵
<Operation>CHARGE</Operation>↵
<IsPost>>true</IsPost>↵
<ExternalNodeId>FLEX</ExternalNodeId>↵
<NodeServiceId>GCPSVC</NodeServiceId>↵
<MdnMsisdn>9495553000</MdnMsisdn>↵
<MonetaryAmount>10</MonetaryAmount>↵
</RatingInvoke>↵
</GapMessage>↵
END
```

Figure 1-2 Sample GAP-Formatted Message with Header and Trailer

## Copying the Charging Requests

Service providers generally use FTP to copy the charging requests to the designated subdirectory in Product. Contact your carrier for the FTP user name, password, and subdirectory location.

## Processing File-Based Interface Files

Company has established subdirectories on the Product SCP-D where the file processing takes place. The names of the subdirectories are configured in the `data_rating_entities.xml` file.

Table 1-1: File Processing Locations

Subdirectory	Contains	Activity
/\$PPHOME/extracts/gcpFtp/read	Files containing GAP-formatted messages (chargingMsgMMDDYYHHMMSS.in)	Service Provider copies the initial charging requests to this subdirectory.
		Product reads the messages and sends them to the GCE for charging.
/\$PPHOME/extracts/gcpFtp/write	Response files (chargingMsgMMDDYYHHMMSS.out) <sup>a</sup>	Product generates successful and unsuccessful (denied) charging responses in this subdirectory.
		Service Provider retrieves response files for analysis and reconciliation.
/\$PPHOME/extracts/gcpFtp/processed	Processed files	Product moves files whose messages have been sent for charging to this subdirectory from /read.
		Carrier periodically purges the files <sup>b</sup> .

- a. For every **chargingMsgMMDDYYHHMMSS.in** file it reads, the product will generate a corresponding response file **chargingMsgMMDDYYHHMMSS.out**.
- b. Carriers can use the Product File Purge Tool to automatically purge the files in the /processed subdirectory. For help setting up the file purge as a cron job, see the Product Platform Administration Guide or contact Company Technical Support.

## Viewing Response Files

Response files (chargingMsgMMDDYYHHMMSS.out) report successful and unsuccessful (denied) charging activity. Unsuccessful charging responses provide response, denial, and error codes. An unsuccessful charging response would look similar to the following.

---

**Note** The message header is indicated by RATING RESPONSE BEGIN. The message trailer is indicated by END.

---

```

RATING RESPONSE
BEGIN
Operation Indicator           [Charge]
Response Type                 [Critical]
Session ID                    [1]
Transaction ID                [1]
GC Session ID                 [0]
Message Reference ID         [2]
Service ID                    [SMSMO]
Network Element ID           [1.50.1.7]
Invoke Time                   [10/8/2002 19:20:34]
Mobile Directory Number       [9498383000]
IMSI/MIN                      [Not Present]
Language Indicator            [Not Present]
Response Code                  [Error]
Denial Reason                  [Not Present]
Error Cause                    [Service Not Found]
Balance Information            [Not Present]
Monetary Amount                [0.0000]
Usage Units                    [100.0000]
Usage Units Type               [Not Present]
Alt Usage Units                [Not Present]
Alt Usage Units Type           [Not Present]
Account Balance                [0.0000]
Funds Expiry                  [10/30/2005 23:59:59]
Text Response                  [Not Present]
Special Billing Response Info  [Not Present]
Announcement ID                [Not Present]
Error Text                     [Not Present]
AccountStatusWarning:         [Not Present]
Unit Of Measure Code          []
Reservation Operation           [None]
Reservation ID                  []

END

```

Figure 1-3 Unsuccessful Charging Response

## Investigating Results

It is the service provider's responsibility to investigate and resolve any errors reported in the files. When you encounter unsuccessful charging responses, always follow these steps:

1. Is there a Response Code of Error?
2. If there is an Error, what is the Denial Reason?
3. If there is a Denial Reason, what is the Error Cause?

### Response Codes

Product uses the following response codes in the response files:

- + Unknown
- + Allowed
- + Allowed with Announcement
- + Denied
- + Denied with Announcement
- + Error
- + Error with Announcement

### Denial Reasons

Product uses the following denial reasons in conjunction with response codes:

- + Unknown
- + Zero Balance
- + Insufficient Funds
- + Inactive Account
- + Blocked

### Error Causes

Product uses the following error causes in conjunction with response codes and denial reasons:

Error Cause	Definition
None	Charge was successful.
Resource Shortage	Some resources are unavailable so the system cannot charge for events.
Unknown Subscriber	Subscriber's IMSI, MDN, or MIN were not found in Product DB.
System Failure	Database error. Cannot connect to the Product DB.
Processing Error	Internal processing error (either memory allocation or exceptions).
Service Not Found	Requested service not found in the Product DB.
Service Not Subscribed	Subscriber tried to invoke a service or Product received an event with service that the subscriber has not subscribed to.

<b>Error Cause</b>	<b>Definition</b>
Operation Mismatch	Invoked operation is not compatible with the service it was invoked for.
Inactive Accounts	Request was received specifying charging on an inactive account (for example, account status of Cooling).
Authorized Usage Not Configured	Default configured usage is requested, but the default value is not configured.
Invalid Parameter	Error occurred in the GCE extracting parameters from the request.
Units Type Not Found	External units ID provided in a UsageUnitsType parameter does not match any configured value.

## Glossary of Terms

The following terms are used in this document:

Table 1-2: Glossary

<b>Term</b>	<b>Definition</b>
GAP	General Application Protocol
GCAP	General Charging Application Protocol
GCE	General Charging Engine
GCP	General Charging Protocol
Service Provider	External node that delivers non-voice services such as SMS, MMS, voice mail, games, etc., to prepaid wireless service consumers

