(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
1 May 2003 (01.05.2003)

PCT

(10) International Publication Number
WO 03/036416 A2

(51) International Patent Classification[7]: G06F

(21) International Application Number: PCT/US02/26033

(22) International Filing Date: 15 August 2002 (15.08.2002)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/312,737 15 August 2001 (15.08.2001) US

(71) Applicant: QUALCOMM INCORPORATED [US/US]; 5775 Morehouse Drive, San Diego, CA 92121 (US).

(72) Inventors: MINEAR, Brian; 13704 Fontanelle Place, San Diego, CA 92128 (US). HOREL, Jerry; 6500 Torin Road, Brentwood Bay, British Columbia V8M2H5 (CA). YU, Julie; 48700 Algonquin Court, San Diego, CA 92130 (US). CHMAYTELLI, Mazen; 2913 Denver Street, San Diego, CA 92117 (US). KLEIN, Michelle; 4104 Kerwood Court, San Diego, CA 92130 (US). MEALER, Vicki; 1331 Mountain Park Place, Escondido, CA 92027 (US). OLIVER, Mitchell, B.; 9737 Caminito Suelto, San Diego, CA 92131 (US).

(74) Agents: WADSWORTH, Philip, R. et al.; Qualcomm Incorporated, 5775 Morehouse Drive, San Diego, CA 92121 (US).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, OM, PH, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, UZ, VC, VN, YU, ZA, ZM, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, SK, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *without international search report and to be republished upon receipt of that report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

WO 03/036416 A2

(54) Title: APPLICATION DISTRIBUTION AND BILLING SYSTEM IN A WIRELESS NETWORK

(57) Abstract: Systems and methods consistent with the present disclosure provide for transaction management, including interfaces to communicate between systems, transaction and billing processing, and product negotiation and management. Using an XML interface as a standard interface, simpler and more efficient communication is provided between systems. Transaction and billing processing aspects provide systems and methods for tracking, processing and managing transactions associated with data on a wireless device. Product negotiation and management aspects provide systems and methods to track data and negotiate data pricing and other metadata between buyers and seller and providers.

APPLICATION DISTRIBUTION AND BILLING SYSTEM IN A WIRELESS NETWORK

CROSS-REFERENCE TO RELATED APPLICATION

This application claims the benefit of Provisional Application No. 60/312,737, filed August 15, 2001, pending, which application is incorporated herein by reference in its entirety.

BACKGROUND OF THE INVENTION

I. Field of the Invention

The present invention generally relates to data networks and computer communications and processing. More particularly, the invention relates to the interfacing between systems, transaction processing and billing, and product negotiation and management.

II. Description of the Related Art

Wireless devices, such as cellular telephones, personal digital assistants ("PDAs"), pagers, laptops with wireless connectivity, etc., communicate packets including voice and data over a wireless network. These wireless devices have installed application programming interfaces ("APIs") onto their local computer platform that allow software developers to create software applications that operate on the wireless device. The API sits between the wireless device system software and the software application, making the wireless device functionality available to the application without requiring the software developer to have the specific wireless device system source code.

The software applications can come pre-loaded at the time the wireless telephone is manufactured, or the user may later request that additional programs be downloaded over cellular telecommunication carrier networks, where the downloaded applications are executable on the wireless telephone. As a result, users of wireless telephones can customize their wireless telephones through the selective downloading of applications, such as games, printed media, stock updates, news, or any other type of information or application that is available for download through the wireless network. In order to manage the cellular telephone resources, the user of the wireless telephone purposefully deletes applications and data from the wireless telephone platform to clear storage space so that new applications can be loaded onto the cleared storage.

In contrast to the larger computer platforms of personal computers and PDAs, wireless devices have limited resources, such as storage and processing, to devote to non-essential applications. Typically, the telecommunication applications have priority of usage of the system resources, with other applications allocated resources as available. The wireless device thus only

has a limited capacity for holding all files for applications, and the managing of resources is left up to the discretion of user of the telephone to delete applications to make room for new applications desired downloaded to the wireless device. The wireless device will not otherwise download an application that it does not have the resources to hold and execute.

Applications, and other data, that will be downloaded to a wireless device will require billing processing. Downloading applications, content or other transactions that occur with a wireless device take up resources on a network. A carrier, in the case of a wireless network, will want to record these transactions and bill for it appropriately.

In the case with voice, a carrier only needs to keep track of the amount of time the wireless device is used on the network and bill for the minutes of use. With data, however, the billing paradigm may be different. Carriers may bill for the download or use of a data application separate from how much time it takes on the carrier's network to download the application. To bill for these transactions, the specific transaction will need to be accounted for and billed, not just the amount of time used on the network to perform the transaction.

In addition, with applications, there may be multiple party settlements involved who share in the fee for the billed transaction. For example with an application download transaction, a carrier and a developer may share the download transaction fee incurred by the wireless device. In other cases, such as with downloading content, the carrier, a content provider and/or a third party involved may get part of the fee incurred by the wireless device's use of that content. Consequently, tracking, billing, and maintaining who shares in the fee for the multitude of transactions that occur becomes quite complex. This becomes even more complex when an extremely high number of transactions that may occur on a carrier's network, with thousands, if not millions, of wireless devices performing numerous transactions each.

Additionally, wireless devices typically need to communicate with other systems and databases within the other systems. Unfortunately, the wireless device may lose a signal during communication or be otherwise unavailable when transmitting information to other systems. This may cause errors when attempting to access a database when the signal is lost. The wireless device may be required to reinitiate the database access and resubmit the database request when the signal is reacquired.

Extended beyond wireless devices, often wire-based systems need to communicate with each other but do not share a similar "language" for communication. For example, as with wireless devices, one system may need to communicate with the database in another system to receive or insert data. To communicate with the database, the system must be aware of the database language, record and field structures, and formats in order to access and store information in the database. While current technologies provide for the ability for the interface

between the system and the database to include the language, structure and format of the database, this becomes more complex when multiple databases, possibly requiring multiple unique database languages, etc., need to be accessed.

Furthermore, when data is to be sent to multiple systems, or conversely received from multiple systems, a common interface does not exist to integrate across the multiple systems to simplify the data transmission. This is problematic for systems communicating with several other systems.

Current methods in the art do not address this need. Database replication services and custom built database interfaces can become very complex and unwieldy if multiple databases need to be accessed. In addition, all systems that access the custom database must have that interface. Also, custom built databases are required to stay online for transactions to occur.

Electronic Data Interchange (EDI) only addresses the need for pre-defined message types and content specific to electronic commerce. The EDI message formats do not address the data exchange needs as it relates to wireless services, system integration considerations and billing specificity.

In addition, products offered for use by the wireless device require negotiation between parties who created the product and the carriers who's networks will transmit the product to the wireless devices. As the product list and number of carriers and wireless devices capable of using the product increases, the negotiation of the product parameters, such as prices, becomes more complex and the managing of all the product offerings and agreed to prices become much more difficult.

Therefore what is needed in the art are systems and methods to address the above recognized shortcomings in the art.

SUMMARY OF THE INVENTION

Systems and methods consistent with the present invention overcome the shortcomings of existing systems by providing systems and methods for interfacing between systems, transaction processing and billing, and product negotiation and management.

In one embodiment of the present invention, a method comprises a method as disclosed herein.

In another embodiment of the present invention, an apparatus comprises an apparatus as disclosed here.

Other objects, advantages, and features of the present invention will become apparent after review of the hereinafter set forth Brief Description of the Drawings, Detailed Description of the Invention, and the Claims.

Table of Contents

Decentralized Transaction Manager

Purpose

The purpose of this concept paper is to describe the distributed Transaction Manager (TXN) implementation approach. It is a high level conceptual description to be used by the QC implementation team and shared with customers to communicate design concepts and assumptions.

Assumptions

The following are implementation assumptions relating to functional requirements and design considerations for decentralized Transaction Manager.

- A carrier may be configured with one of the following TXN configurations
  a) The baseline Transaction Manager configuration, where transactions are consolidated in a Master TXN (MTXN) located in San Diego with network connectivity between the carrier ADS farms and QC, San Diego.
  b) A distributed Carrier TXN (CTXN) where the TXN host and services are located at a carrier data center.
- If the carrier selects a distributed TXN configuration, QC will provide the carrier with a parts list that includes required hardware and software for the carrier to purchase. The specifics of the parts list shall be determined based upon the carrier's sizing, reliability and availability requirements.
- Oracle RDBMS will be required to be used as the repository for both baseline and decentralized TXN configurations.
- All changes to the TXN data model will be made through the QC software release process. This applies to both the CTXN and MTXN. Carriers will not be able to modify the data model independent of the QC baseline configuration. QC will routinely and on-demand perform and checksum/hash of installed TXN components to confirm integrity of TXN released configuration. This functionality is required to provide revenue assurance for both Carrier and QUALCOMM.
- In a decentralized TXN configuration, the carrier is responsible for routine server and database maintenance. This includes system monitoring, backup/recovery, and escalation of support to QC as specified according to carrier agreements.

Functional Architecture

Centralized TXN Configuration (Baseline)

Figure 1 depicts the Centralized TXN configuration. The following are key functional characteristics of the Centralized TXN configuration.

- (1) The carrier ADS Farm uploads raw phone transactions into MTXN.

- (2) Business metadata is replicated from UAM to MTXN for transaction rating and conversion.
- (3) All carrier transactions are consolidated, converted and rated by MTXN.
- (3) MTXN is a centralized Oracle repository located in a QC data center.
- (3) Carrier transactions are logically separated but physically consolidated in MTXN, an Oracle repository.
- (4) Authorized Carrier users utilize the carrier extranet to apply adjustments to applications and generate usage reports by accessing MTXN.
- (5) Carriers may send requests to QC to apply additional application adjustments (ISV adjustments not mapping to transactions, recall, etc.).
- (6) Carrier billing extract files, in the form of XML file extracts, will be generated periodically (i.e., every 30 minutes) and sent to the carrier for consumer billing. The carrier billing extract files include standard and restricted application transactions along will all adjustments and MIN update events (i.e., MIN transfer, MIN deactivate).
- (7) Carriers perform other consumer adjustments in their billing system(s).
- (8) According to carrier agreements, carriers may be required to self-report on BREW enablement fees.
- (9) BREW Billing processes the MTXN rated transactions and derives carrier invoices, according to carrier agreements.
- (9) BREW Billing processes the MTXN rated transactions and processes ISV payment according to ISV agreements.
- (10) Authorized carrier users utilize the carrier extranet to access billing detail information by accessing BREW billing.

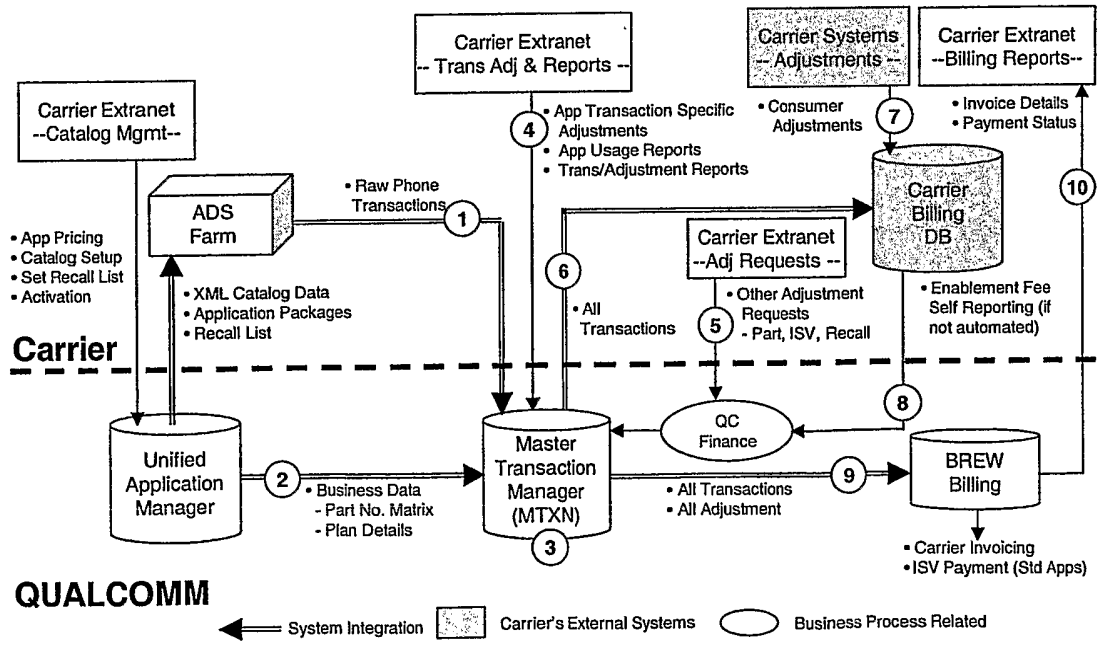


Figure 1 – Baseline Centralized Transaction Manager

Distributed TXN Configuration

Figure 2 depicts the Distributed TXN configuration. The following are key functional characteristics of the Distributed TXN configuration.

- (1) The carrier ADS Farm uploads raw phone transactions into a Carrier Transaction Manager (CTXN).
- (2) Business metadata is replicated from UAM to CTXN and Master Transaction Manager (MTXN) for transaction rating and conversion.
- (3) All carrier transactions are consolidated, converted and rated by CTXN.
- (3) CTXN is an Oracle repository located in a Carrier data center.
- (3) Carrier transactions are logically and physically separated in a CTXN Oracle repository.
- (4) All transactions and transaction adjustments are replicated to MTXN.
- (5) MTXN is a centralized Oracle repository located in a QC data center.
- (5) Carrier transactions are logically separated physically consolidated in MTXN.
- (6) Authorized Carrier users utilize the carrier extranet to apply transaction adjustments to applications and generate usage reports by accessing CTXN.
- (7) Carriers may send requests to QC to apply additional standard application adjustments (by Part Number or ISV) that do not mapping to specific transaction, or request a recall adjustment pertaining to either a standard or restricted application in MTXN.
- (8) Adjustments that are not associated with a transaction are applied by QC in MTXN and get propagated to CTXN.
- (9) Carrier configures integration between CTXN and their billing system either with (a) XML billing extract files generated periodically (e.g., every 10 minutes) or (b) CTXN API call for near real-time data flow. This data is used in carrier billing system for consumer billing and includes standard and restricted application transactions along will all adjustments and MIN update events (i.e., MIN transfer, MIN deactivate).
- (10) Carriers perform other consumer adjustments in their billing systems.
- (11) According to carrier agreements, carriers may be required to self-report on enablement fees.
- (12) BREW Billing processes the rated transactions in MTXN and derives carrier invoices, according to carrier agreements.
- (12) BREW Billing processes the MTXN rated transactions and processes ISV payment according to ISV agreements.
- (13) Authorized carrier users utilize the carrier extranet to access billing detail information by accessing BREW billing.

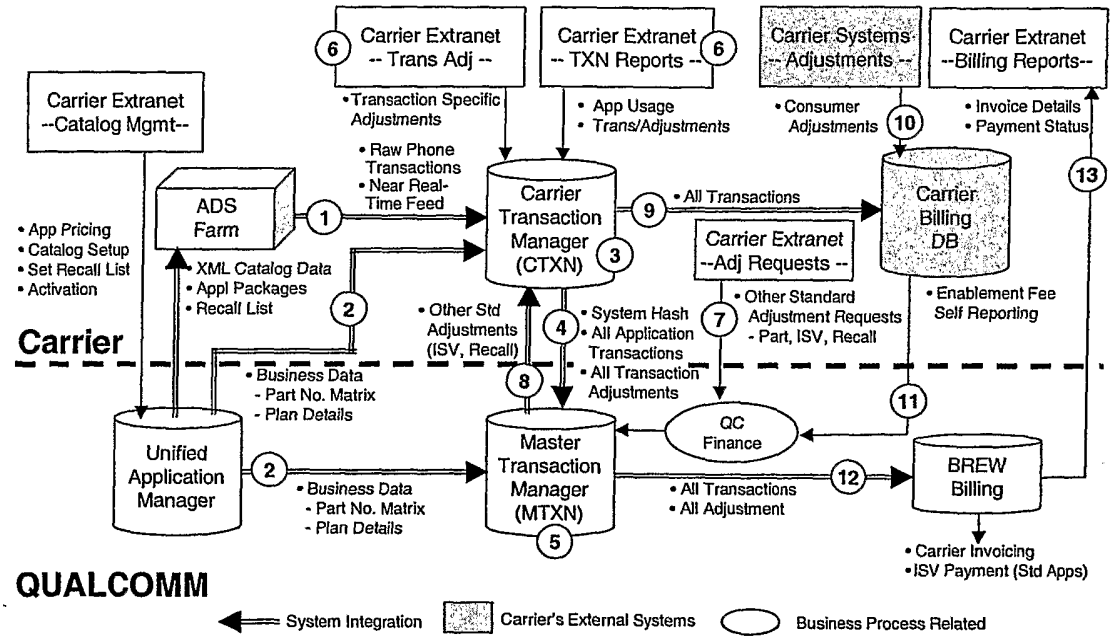


Figure 2 – Distributed Transaction Manager

Data Interface Considerations

Figure 4 shows the interfaces between QIS Middleware systems. Method of data transfer between the systems is also identified.

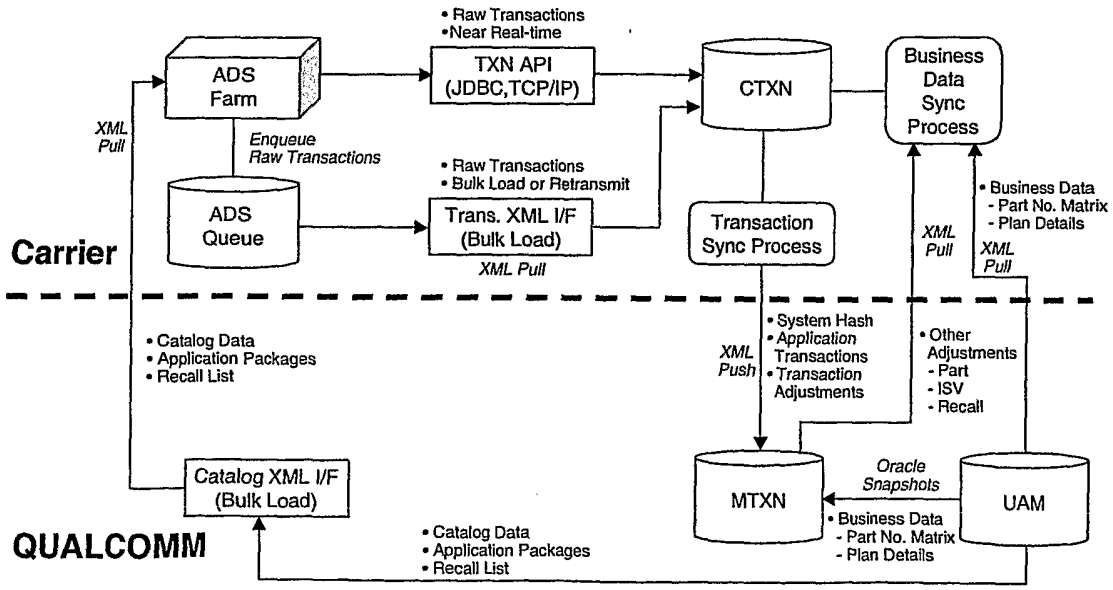


Figure 4 – Distributed Transaction Manager

Performance Considerations

This section is TBD.

Example Decentralized TXN Parts List

This section is TBD.

Distributed Transaction Manager – Concept Paper

Revision: D

Print Date: 8/7/01

QUALCOMM Incorporated
5775 Morehouse Drive
San Diego, CA 92121-1714

Table of Contents

Decentralized Transaction Manager

Purpose

The purpose of this concept paper is to describe the distributed Transaction Manager (TXN) implementation approach. It is a high level conceptual description to be used by the QC implementation team and shared with customers to communicate design concepts and assumptions.

Assumptions

The following are implementation assumptions relating to functional requirements and design considerations for decentralized Transaction Manager.

- A carrier may be configured with one of the following TXN configurations
  a) The baseline Transaction Manager configuration, where transactions are consolidated in a Master TXN (MTXN) located in San Diego with network connectivity between the carrier ADS farms and QC, San Diego.
  b) A distributed Carrier TXN (CTXN) where the TXN host and services are located at a carrier data center.
- If the carrier selects a distributed TXN configuration, QC will provide the carrier with a parts list that includes required hardware and software for the carrier to purchase. The specifics of the parts list shall be determined based upon the carrier's sizing, reliability and availability requirements.
- Oracle RDBMS will be required to be used as the repository for both baseline and decentralized TXN configurations.
- All changes to the TXN data model will be made through the QC software release process. This applies to both the CTXN and MTXN. Carriers will not be able to modify the data model independent of the QC baseline configuration. QC will routinely and on-demand perform and checksum/hash of installed TXN components to confirm integrity of TXN released configuration. This functionality is required to provide revenue assurance for both Carrier and QUALCOMM.
- In a decentralized TXN configuration, the carrier is responsible for routine server and database maintenance. This includes system monitoring, backup/recovery, and escalation of support to QC as specified according to carrier agreements.

Restricted Application Handling

The BREW business model supports Standard Applications that use BREW processes provided by QUALCOMM for certification, pricing, and developer payment; and Restricted Applications where the Carrier is responsible for testing the applications, submitting them to QUALCOMM and paying the developer. The QIS Middleware can support two different approaches for handling Restricted Application billing in decentralized Transaction Manager.

- **Automatic Reporting** – The Carrier would utilize the QIS Middleware, specifically CTXN, to enter transaction adjustments for Restricted Applications. All transaction records would transfer to MTXN allowing the Carrier to use consolidated billing reports provided by QIS Middleware.

- **Self Reporting** – The Carrier would utilize their existing billing system to make adjustments to Restricted Application transactions and would self report to QUALCOMM data pertaining to Restricted Applications. QUALCOMM and the Carrier would have to negotiate the format of this report and method of delivery.

Each carrier should assess the pros and cons of each approach, before determining preferred method. There may be business terms associated with each option that are not covered in this document. This document will only cover the functional differences between the two alternatives.

Functional Architecture

Centralized TXN Configuration (Baseline)

Figure 1 depicts the Centralized TXN configuration. The following are key functional characteristics of the Centralized TXN configuration.

- (1) The carrier ADS Farm uploads raw phone transactions into MTXN.
- (2) Business metadata is replicated from UAM to MTXN for transaction rating and conversion.
- (3) All carrier transactions are consolidated, converted and rated by MTXN.
- (3) MTXN is a centralized Oracle repository located in a QC data center.
- (3) Carrier transactions are logically separated but physically consolidated in MTXN, an Oracle repository.
- (4) Authorized Carrier users utilize the carrier extranet to apply adjustments to applications and generate usage reports by accessing MTXN.
- (5) Carriers may send requests to QC to apply additional application adjustments (ISV adjustments not mapping to transactions, recall, etc.).
- (6) Carrier billing extract files, in the form of XML file extracts, will be generated periodically (i.e, every 30 minutes) and sent to the carrier for consumer billing. The carrier billing extract files include standard and restricted application transactions along will all adjustments and MIN update events (i.e., MIN transfer, MIN deactivate).
- (7) Carriers perform other consumer adjustments in their billing system(s).
- (8) According to carrier agreements, carriers may be required to self-report on BREW enablement fees.
- (9) BREW Billing processes the MTXN rated transactions and derives carrier invoices, according to carrier agreements.
- (9) BREW Billing processes the MTXN rated transactions and processes ISV payment according to ISV agreements.
- (10) Authorized carrier users utilize the carrier extranet to access billing detail information by accessing BREW billing.

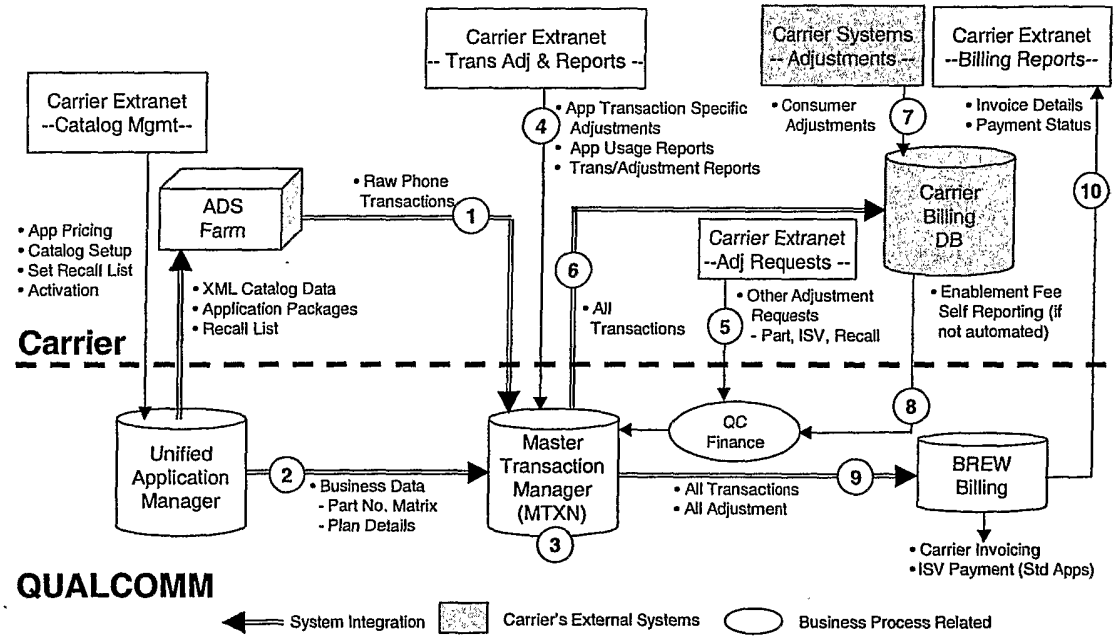


Figure 1 – Baseline Centralized Transaction Manager

Distributed TXN Configuration – Automatic Reporting

Figure 2 depicts the Distributed TXN configuration with automatic reporting. The following are key functional characteristics of the Distributed TXN configuration.

- (1) The carrier ADS Farm uploads raw phone transactions into a Carrier Transaction Manager (CTXN).
- (2) Business metadata is replicated from UAM to CTXN and Master Transaction Manager (MTXN) for transaction rating and conversion.
- (3) All carrier transactions are consolidated, converted and rated by CTXN.
- (3) CTXN is an Oracle repository located in a Carrier data center.
- (3) Carrier transactions are logically and physically separated in a CTXN Oracle repository.
- (4) All transactions and transaction adjustments are replicated to MTXN.
- (5) MTXN is a centralized Oracle repository located in a QC data center.
- (5) Carrier transactions are logically separated physically consolidated in MTXN.
- (6) Authorized Carrier users utilize the carrier extranet to apply transaction adjustments to applications and generate usage reports by accessing CTXN.
- (7) Carriers may send requests to QC to apply additional standard application adjustments (by Part Number or ISV) that do not mapping to specific transaction, or request a recall adjustment pertaining to either a standard or restricted application in MTXN.
- (8) Adjustments that are not associated with a transaction are applied by QC in MTXN and get propagated to CTXN.

- (9) Carrier configures integration between CTXN and their billing system either with (a) XML billing extract files generated periodically (e.g., every 10 minutes) or (b) CTXN API call for near real-time data flow. This data is used in carrier billing system for consumer billing and includes standard and restricted application transactions along will all adjustments and MIN update events (i.e., MIN transfer, MIN deactivate).
- (10) Carriers perform other consumer adjustments in their billing systems.
- (11) According to carrier agreements, carriers may be required to self-report on enablement fees.
- (12) BREW Billing processes the rated transactions in MTXN and derives carrier invoices, according to carrier agreements.
- (12) BREW Billing processes the MTXN rated transactions and processes ISV payment according to ISV agreements.
- (13) Authorized carrier users utilize the carrier extranet to access billing detail information by accessing BREW billing.

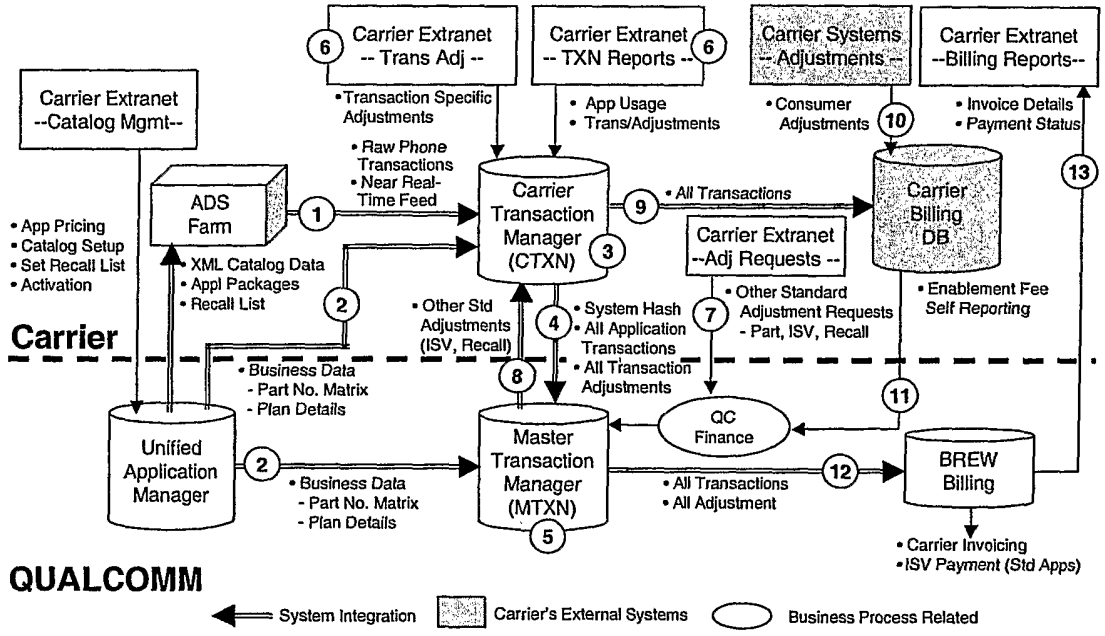


Figure 2 – Distributed Transaction Manager (Auto-Reporting)

Distributed TXN Configuration – Self Reporting

Figure 3 depicts the Distributed TXN configuration with self-reporting. The following are key functional characteristics of the Distributed TXN configuration. Those that changed for self-reporting are highlighted in **bold.**

- (1) The carrier ADS Farm uploads raw phone transactions into a Carrier Transaction Manager (CTXN).

- (2) Business metadata is replicated from UAM to CTXN and Master Transaction Manager (MTXN) for transaction rating and conversion.
- (3) All carrier transactions are consolidated, converted and rated by CTXN.
- (3) CTXN is an Oracle repository located in a Carrier data center.
- (3) Carrier transactions are logically and physically separated in a CTXN Oracle repository.
- **(4) Standard Application transactions and Standard Application transaction adjustments are replicated to MTXN.**
- (5) MTXN is a centralized Oracle repository located in a QC data center.
- (5) Carrier transactions are logically separated physically consolidated in MTXN.
- **(6) Authorized Carrier users utilize the carrier extranet to apply transaction adjustments to Standard Applications and generate usage reports by accessing CTXN.**
- (7) Carriers may send requests to QC to apply additional standard application adjustments (by Part Number or ISV) that do not mapping to specific transaction, or request a recall adjustment pertaining to either a standard or restricted application in MTXN.
- (8) Adjustments that are not associated with a transaction are applied by QC in MTXN and get propagated to CTXN.
- (9) Carrier configures integration between CTXN and their billing system either with (a) XML billing extract files generated periodically (e.g., every 10 minutes) or (b) CTXN API call for near real-time data flow. **This data is used in carrier billing system for consumer billing and includes standard and restricted application transactions along will Standard Application adjustments and MIN update events (i.e., MIN transfer, MIN deactivate).**
- **(10) Carriers perform other consumer adjustments and restricted application adjustments in their external billing systems.**
- **(11) According to carrier agreements, carriers may be required to self-report on enablement fees and application usage.**
- (12) BREW Billing processes the rated transactions in MTXN and derives carrier invoices, according to carrier agreements.
- (12) BREW Billing processes the MTXN rated transactions and processes ISV payment according to ISV agreements.
- (13) Authorized carrier users utilize the carrier extranet to access billing detail information by accessing BREW billing.

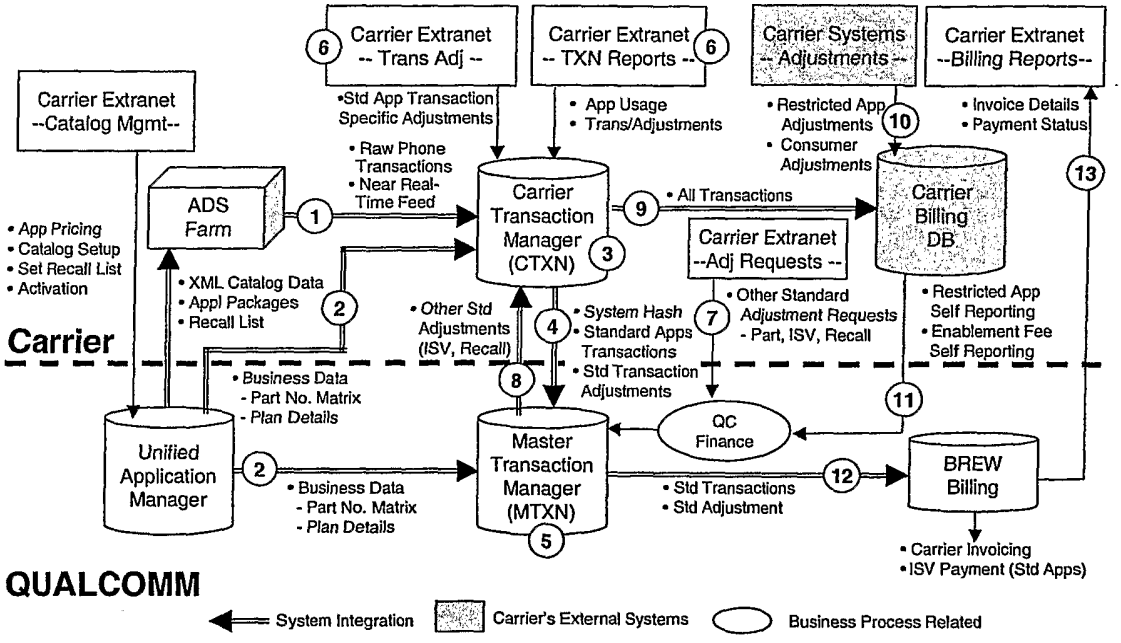


Figure 3 – Distributed Transaction Manager (Self-Reporting)

Data Interface Considerations

Figure 4 shows the interfaces between QIS Middleware systems. Method of data transfer between the systems is also identified.

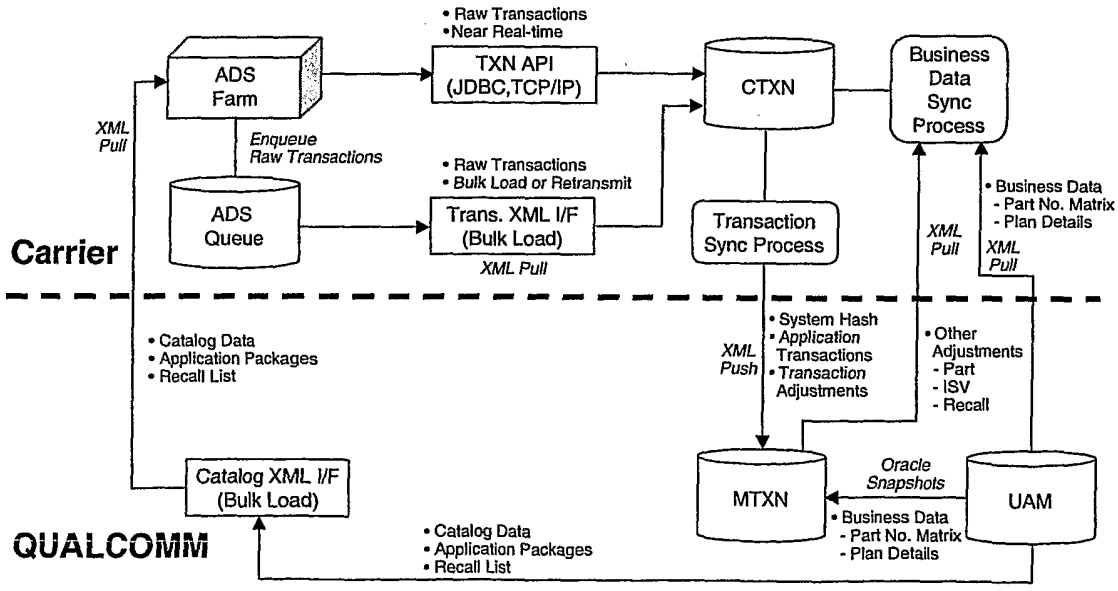


Figure 4 – Distributed Transaction Manager Data Interfaces

Performance Considerations

This section is TBD.

Example Decentralized TXN Parts List

This section is TBD.

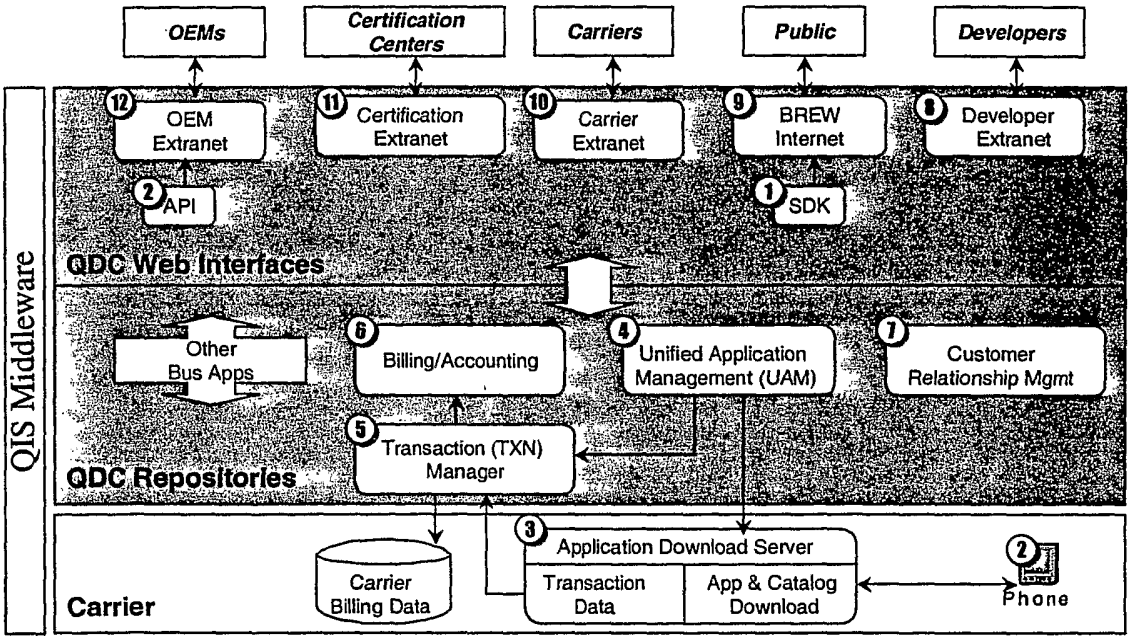
Table of Contents

Transaction Manager

System Overview

The QIS Distribution Center (QDC) provides applications and services relating to the QIS as depicted in Figure 1.

Figure 1. QIS Middleware Conceptual Architecture



These QDC applications and services include the following functions:

OEM Extranet (12) – The OEMs will be provided extranet services to assist them in provisioning the wireless devices with the BREW API, MobileShop, BREW applications and other BREW required components. The OEM extranet will also enable the OEMs to request the creation of additional BREW platform IDs and to submit Restricted Applications for signing and packaging

Certification Extranet (11) – The Certification extranet provides services which facilitate communication between the BREW certification centers (initially NSTL) and the Certification Center headquarters in San Diego. Services provided via the extranet

include: a) ESN management functions for generating test signatures for applications to be executable on test phones; b) catalog management services for managing the certification center catalogs; c) certification status management for tracking whether an application's certification status is submitted, certified or failed; d) access to certification metric reports; and e) access to applications and documentation.

Carrier Extranet (10, 4) – The carrier uses the carrier extranet application service to manage the carrier catalog which contains the selected BREW applications to distribute to their consumers. Applications selected by the carrier for distribution will be replicated to the Carrier ADS (3) along with the carrier catalog updates. The phone (2) will interact with the Carrier ADS (3) to download the application to the phone and activate the application for the phone user. Phone transactions are logged at the ADS and uploaded to the transaction manager for Billing conversion and rating (3,5). In addition, the carrier extranet provides billing support services that augments the invoicing and financial reconciliation process with QIS and carriers. The carrier extranet will also contain facilities for submitting and processing of Restricted Applications from the carrier.

BREW Internet (9) – The BREW internet pages will be available to the public off of the QC internet site. The pages are targeted to provide general BREW information to the public, developers, OEMs, and carriers. General BREW marketing information, white papers, technical information etc, will be available on the internet. In addition, developers will be able to download the BREW SDK and access to basic technical information relating to BREW and the SDK.

Developer Extranet (8) – The Developer Extranet provides BREW services to the Independent Software Vendor (ISV). Included in these services is the ability for the ISV to maintain the application price plans and access to ISV billing support services. In addition, there is access to various BREW development tools, documents (i.e., Carrier Guidelines), access to certification center, and the procedures for a developer company to sign on as a BREW ISV.

Unified Application Management (UAM) (4) – UAM is a repository which will manage certified and pre-certified applications as well as carrier catalogs. Once an application is certified by the Certification Center, the ACCHQ sets the application "ready for distribution" to the UAM via the Certification Extranet (11). In addition, Restricted Applications enter the UAM after going through ACCHQ only for signing. Once the application is marked ready for distribution the application is made available for carrier distribution via UAM (4) and the Carrier Application Catalog Management functions on the Carrier Extranet (10).

Transaction Manager (TXN) (5) – Transaction Manager is a data repository which receives uploads of phone transaction data from multiple ADS farms. TXN consolidates the raw phone transactions and then processes the data by converting information and applying pricing to each transaction (i.e., a rating process). Once the transactions are converted and rated, Billing and Accounting programs process the information and prepare to process financial transactions (6). These financial transactions include carrier invoicing and developer payment. TXN is also used as the primary repository for applying adjustments and generating Carrier Billing Extract files to enable the carrier to bill their phone users for application purchases.

Customer Relationship Management (CRM) Services (6) – Carriers, OEMs, Developers, Customer Prospects and Certification Centers will be provided customer

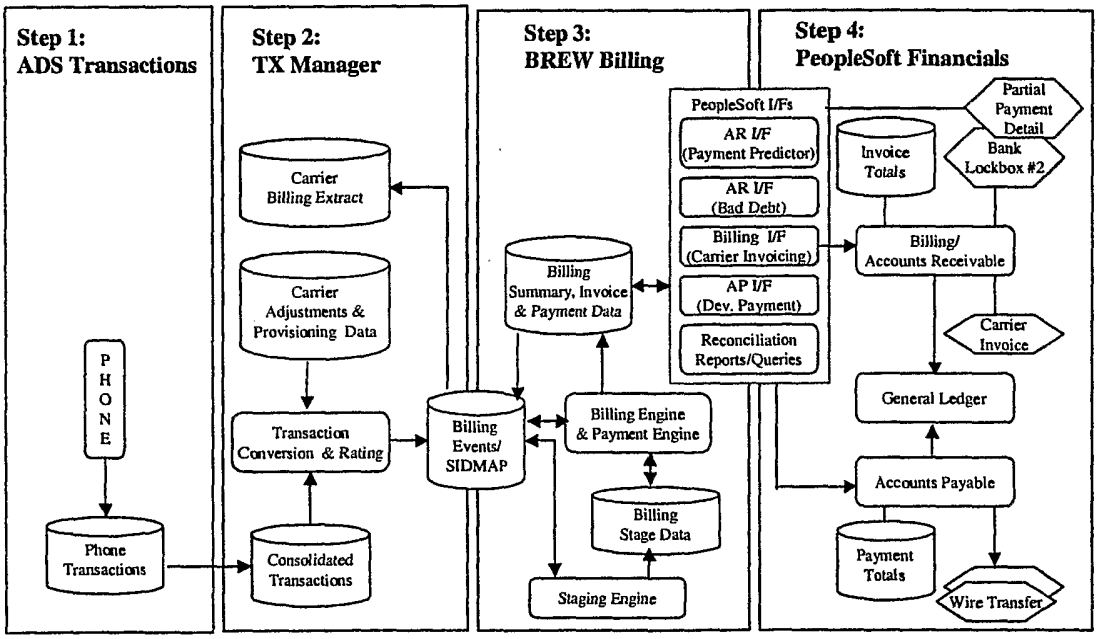
support services. Customer tickets will be managed via the CRM system and an escalation path will be implemented for addressing customer support requests.

Billing/Accounting Services (6) – As part of the distribution services QC will provide accounting "clearing house" services for the primary carriers and developers. As such, the QDC will collect phone transaction information which will be used in conjunction with carrier provisioning data and accounting adjustments to compile a set of transactions as the foundation for invoicing the primary carrier. This invoice shall include QC fees and developer fees based on application usage and according to the carrier agreements.

Transaction Manager Overview

This section describes the Transaction Manager functions as depicted in the Figure below.

TX/Billing Data Flow Diagram



Transaction History

The Transaction Manager will maintain the consolidated transaction history from all carrier ADS. The ADS will create the transaction logs which will be periodically loaded into the consolidated ads transaction history.

The CRM system will be provided with access into the transaction history for customer support. Transactions will be archived to offline storage on a periodic basis.

Carrier ADS Transaction Log will include more transaction records than those needed for billing purposes (i.e., interim download events). These additional transaction records may be required for system troubleshooting and customer support. The transaction history repository will maintain security such that each carrier's data is logically separated from every other carrier's data.

The phone will transmit a subscriber id with each transmission. For most carriers this subscriber id is simply the phone number of the phone. For some carriers this will be a number that is meaningful only to the carrier that uniquely identifies a customer.

The events that the ADS will provide the Transaction manager are

a) Download Acknowledgments - This event has two sub types. The initial download acknowledgment and in the case of a failed reply a subsequent download acknowledgement.

b) Application deletion – This event is sent when an application is deleted

c) Debug – These events are for debug purposes only and are filtered out and not inserted into transaction repository

d) Monitor – These events are text strings used for monitoring ADS statistics.

e) Other – These events are text strings that can be used for application specific events.

Provisioning Events

The Transaction Manager repository will provide a mechanism to insert external phone provisioning events and subscriber(customer) data into the transaction repository. This provides the system with the means to account for preinstalled applications and applications/subscription that are transferred from one phone to another with the same customer. A pseudo customer identifier identifies customers. This identifier is an internal key that associates subscriber id's (phone numbers) together. When a customer changes subscriber id (phone number) the system maintains the relationship between the applications/subscriptions the customer had on the old subscriber id(phone number) with the new subscriber id (phone number).

Provisioning events are provided to the Transaction Manager as a XML file. The events contained in the file are

a) Subscriber deactivation – This event is sent when a subscriber/MIN is deactivated

b) Subscriber transfer – This event is sent when a subscriber is given a new subscriber id (phone number)

Billing Data Creation/Extraction

The Transaction Manager repository will convert the raw ADS transaction data and provisioning data into a composite billing data set. The transaction manager will filter out all non-billing related data. It will convert internal application identifiers to part numbers, convert pricing handles into actual price information and convert time stamps expressed as IS-95 time (seconds since 0:0:0 Jan 6 1980) to local time and GMT.

It will also provide a mechanism to add adjustments to the data. The data will be stored in a composite billing data set and provided to the billing system for processing into the various invoices required. The Transaction Manager will provide a mechanism to extract this information into XML files for transmission to the carrier for the purposes of billing consumers and in some cases providing payment to developers.

2 Transaction Manager Interfaces

This section identifies the Transaction Manager Document Type Definitions (DTDs) that describe the XML file formats for carrier and TX manager data exchange.

Provisioning Data

```
<!--XML DTD for Carrier Provisioning Data -->
<!-- version $Revision: #9 $ -->
<!-- provides mechanism for notification of Subscriber id changes -->

<!ELEMENT EXTEVENT (CARRIERNAME,GMT,DID,TX*,CT) >
<!ELEMENT CARRIERNAME (#PCDATA)>          <!-- Primary carrier Name -->
<!ELEMENT GMT (#PCDATA)>                  <!-- GMT time format MMDDYYYY HHMISS -->
<!ELEMENT DID (#PCDATA)>                  <!-- Document id -->

<!ELEMENT TX (EV+) >                      <!-- transaction boundary -->

<!ELEMENT EV (SID,CID?,TY,TS,DA*) >       <!-- event -->

<!ELEMENT SID (#PCDATA)>                  <!-- subscriber id generating event -->
<!ELEMENT CID (#PCDATA)>                  <!-- carrier id from phone -->
<!ELEMENT TY (#PCDATA)>                   <!-- event type -->
<!ELEMENT TS (#PCDATA)>                   <!-- event timestamp -->

<!ELEMENT DA (OT*,MI*)>                   <!-- event data -->
<!ELEMENT OT (#PCDATA)>                   <!-- Other Event data -->

<!ELEMENT MI (ACT,NSID?)>                 <!-- min event -->
<!ELEMENT ACT (#PCDATA)>                  <!-- unactivate, transfer-->
<!ELEMENT NSID (#PCDATA)>                 <!-- new subscriber id assigned -->

<!ELEMENT CT (#PCDATA)>                   <!—record count -->
```

Provisioning Data XML Data Mapping

<!--XML DTD for Carrier Provisioning Data --> **<!-- version $Revision: #1 $ -->**	**DESCRIPTION**	**TXN DATA FORMAT**	**VALID FOR**	**NOTES**
<!ELEMENT EXTEVENT (CARRIERNAME,GMT,DID,TX*, CT) >				
<!ELEMENT CARRIERNAME (#PCDATA)>	<!—Primary carrier name -->	30an	all	
<!ELEMENT GMT (#PCDATA)>	<!-- GMT time format MMDDYYYY HHMISS -->	MMDDYYYY HHMISS	all	
<!ELEMENT DID (#PCDATA)>	<!-- unique document ID -->	22an	all	*DID is a unique document ID that identifies a specific carrier provisioning load file.*
<!ELEMENT TX (EV+) >	<!-- transaction boundary -->			
<!ELEMENT EV (SID,CID?,TY,TS,DA*) >	<!-- event -->			
<!ELEMENT SID (#PCDATA)>	<!-- subscriber id generating event -->	200an	TR, DA	note format of SID is determined by what the carrier provisions on the phone.
<!ELEMENT CID (#PCDATA)>	<!-- carrier id from phone -->	10n	TR, DA	Numeric value, 10n maximum, not padded with leading zeros
<!ELEMENT TY (#PCDATA)>	<!-- event type -->	2an	TR, DA	"MI" for MIN Update
<!ELEMENT TS (#PCDATA)>	<!-- local time stamp -->	MMDDYYYY HHMISS	TR, DA	
<!ELEMENT DA (OT*,MI*)>	<!-- event data -->			
<!ELEMENT OT (#PCDATA)>	<!-- Other Event data -->	TBD	TR, DA	Extra field if additional information needs to be passed with the event
<!ELEMENT MI (ACT,NSID?)>	<!-- min event -->			
<!ELEMENT ACT (#PCDATA)>	<!—deactivate or transfer-->	5an	TR, DA	"DA" for Subscriber ID deactivation or "TR" for Subscriber ID transfer
<!ELEMENT NSID (#PCDATA)>	<!-- new subscriber id assigned -->	200an	TR	note format of new SID is determined by what the carrier provisions on the phone.
<!ELEMENT CT (#PCDATA)>	<!—record count -->	22n	all	Record count of EV records

Example Provisioning Data XML File

```
<?xml version="1.0" ?>
<!DOCTYPE EXTEVENT (View Source for full doctype...)>
- <EXTEVENT>
    <CARRIERNAME>Qualcomm</CARRIERNAME>
    <GMT>01042001 172246</GMT>
    <DID>1000000</DID>
  - <TX>
    - <EV>
      - <!--
          -->
        <SID>858900000718</SID>
        <CID>1</CID>
        <TY>MI</TY>
        <TS>01012001 172246</TS>
      - <DA>
        - <MI>
            <ACT>DA</ACT>
          </MI>
        </DA>
      </EV>
    - <EV>
      - <!--
          -->
        <SID>858900000720</SID>
        <CID>1</CID>
        <TY>MI</TY>
        <TS>01012001 172246</TS>
      - <DA>
        - <MI>
            <ACT>TR</ACT>
            <NSID>858900000800</NSID>
          </MI>
        </DA>
      </EV>
    </TX>
    <CT>2</CT>
  </EXTEVENT>
```

Adjustment Data

```
<!--XML DTD for Adjustment records  for Bulk Data Load -->
<!-- version $Revision: #1 $ -->

<!ELEMENT ADJUST (CARRIER,CID,DID,GMT,AJ*,CT)>

<!ELEMENT CARRIER (#PCDATA)>        <!-- carrier name -->
<!ELEMENT CID (#PCDATA)>            <!-- sub carrier id -->
<!ELEMENT DID (#PCDATA)>            <!-- unique document ID -->
<!ELEMENT GMT (#PCDATA)>            <!-- GMT time format MMDDYYYY HHMISS -->

<!ELEMENT AJ (EI,AR,AD,AP,AE)>      <!-- Ajustment record -->
<!ELEMENT EI (#PCDATA)>             <!-- event id -->
<!ELEMENT AR (#PCDATA)>             <!-- adjustment reason -->
<!ELEMENT AD (#PCDATA)>             <!-- adjustment dap -->
<!ELEMENT AP (#PCDATA)>             <!-- adjustment price -->
<!ELEMENT AE (#PCDATA)>             <!-- adjustment description -->
<!ELEMENT AI (#PCDATA)>             <!-- adjusted event id -->
<!ELEMENT CT (#PCDATA)>             <!-- record count -->
```

Adjustment Data XML Data Mapping

<!--XML DTD for Adjustment Data Bulk Load -->	DESCRIPTION	TXN DATA FORMAT	VALID FOR	NOTES
<!-- version $Revision: #1 $ -->				
<!ELEMENT ADJUST (CARRIER,CID, DID,GMT,AJ*, CT)>				
<!ELEMENT CARRIER (#PCDATA)>	<!-- carrier name -->	30an	all	
<!ELEMENT CID (#PCDATA)>	<!--sub-carrier ID -->	10n	all	
<!ELEMENT DID (#PCDATA)>	<!-- unique document ID -->	22an	all	DID is a unique document ID that identifies a specific carrier adjustment load file.
<!ELEMENT GMT (#PCDATA)>	<!-- GMT time format MMDDYYYY HHMISS -->	MMDDYYYY HHMISS	all	
<!ELEMENT AJ (EI,AR,AD,AP,AE,AI)>	<!-- Ajustment record -->			
<!ELEMENT EI (#PCDATA)>	<!-- event id -->		TA	Carrier sets this value = 0
<!ELEMENT AR (#PCDATA)>	<!-- adjustment reason -->	2an	TA	1 = Duplicate Download 2 = Accidental Download 3 = Customer Dissatisfaction
<!ELEMENT AD (#PCDATA)>	<!-- adjustment dap -->	S10.2n	TA	sign digit (explicitly +/-), 10 characters, decimal point 2 digits total 14. Must be in U.S. currency
<!ELEMENT AP (#PCDATA)>	<!-- adjustment price -->	S10.2n	TA	sign digit (explicitly +/-), 10 characters, decimal point 2 digits total 14. Must be in the same currency as list price is displayed in catalog
<!ELEMENT AE (#PCDATA)>	<!-- adjustment description -->	240an	TA	Free text description field
<!ELEMENT AI (#PCDATA)>	<!-- adjusted event id -->	12n	TA	Same format as event id, this identifies the original billing event to apply the adjustment to.
<!ELEMENT CT (#PCDATA)>	<!--record count -->	22n	all	Record count of AJ records

Example Adjustment Data XML File

```
<?xml version="1.0" ?>
  <!DOCTYPE ADJUST (View Source for full doctype...)>
- <ADJUST>
    <CARRIER>Qualcomm</CARRIER>
    <CID>1</CID>
    <DID>12345</DID>
    <GMT>01012001 154233</GMT>
  - <AJ>
      <EI>0</EI>
      <AR>3</AR>
      <AD>-4.50</AD>
      <AP>-10.00</AP>
      <AE>this is an example of a customer dissatisfaction
        adjustment</AE>
      <AI>209665</AI>
    </AJ>
  - <AJ>
      <EI>0</EI>
      <AR>1</AR>
      <AD>-6.50</AD>
      <AP>-15.00</AP>
      <AE>this is an example of a duplicate download
        adjustment</AE>
      <AI>209667</AI>
    </AJ>
  - <AJ>
      <EI>0</EI>
      <AR>2</AR>
      <AD>-5.50</AD>
      <AP>-12.00</AP>
      <AE>this is an example of an accidental download
        adjustment</AE>
      <AI>209669</AI>
    </AJ>
    <CT>3</CT>
  </ADJUST>
```

Carrier Billing Extract

The Carrier Billing Extract file is an XML file containing a set of billing transactions provided to the carrier. It is defined with the following DTD.

```
<!--XML DTD for Billing Extract records -->
<!-- version $Revision: #1 $ -->

<!--XML DTD for App Billing -->
<!ELEMENT BILLING (CARRIER,DID,GMT,BR*,CT)>

<!ELEMENT CARRIER (#PCDATA)>        <!-- carrier name -->
<!ELEMENT DID (#PCDATA)>            <!-- unique document ID -->
<!ELEMENT GMT (#PCDATA)>            <!-- GMT time format MMDDYYYY
HHMISS -->

<!ELEMENT BR
(EI,CID,ADS,SID,NSID,PN,NM,MF,SN,RF,LTS,GMT,TY,PM,PB,PV,PR,CY,DP,AR,AD,
AP,AE,AI)> <!-- billing record -->
<!ELEMENT EI (#PCDATA)>         <!-- event id -->
<!ELEMENT CID (#PCDATA)>        <!-- sub carrier id -->
<!ELEMENT ADS (#PCDATA)>        <!-- download server -->
<!ELEMENT SID (#PCDATA)>        <!-- subscriber id (phone number) -->
<!ELEMENT NSID (#PCDATA)>       <!-- new subscriber id (phone number) -->
<!ELEMENT PN (#PCDATA)>         <!-- part number -->
<!ELEMENT NM (#PCDATA)>         <!-- part name -->
<!ELEMENT MF (#PCDATA)>         <!-- manufacturer -->
<!ELEMENT SN (#PCDATA)>         <!-- subscription plan name -->
<!ELEMENT RF (#PCDATA)>         <!-- restricted flag -->
<!ELEMENT LTS (#PCDATA)>        <!-- local time stamp -->
<!ELEMENT TY (#PCDATA)>         <!-- Type -->
<!ELEMENT PM (#PCDATA)>         <!-- price method -->
<!ELEMENT PB (#PCDATA)>         <!-- price basis -->
<!ELEMENT PV (#PCDATA)>         <!-- price value -->
<!ELEMENT PR (#PCDATA)>         <!-- price -->
<!ELEMENT CY (#PCDATA)>         <!-- currency -->
```

```
77    <!ELEMENT DP (#PCDATA)>      <!-- developer price -->
78    <!ELEMENT AR (#PCDATA)>      <!-- adjustment reason -->
79    <!ELEMENT AD (#PCDATA)>      <!-- adjustment dap -->
80    <!ELEMENT AP (#PCDATA)>      <!-- adjustment price -->
81    <!ELEMENT AE (#PCDATA)>      <!-- adjustment description -->
82    <!ELEMENT AI (#PCDATA)>      <!-- adjusted event id -->
83    <!ELEMENT CT (#PCDATA)>      <!-- record count -->
```

Carrier Billing Extract XML Data Mapping

<!--XML DTD for Billing Extract records --> <!-- version $Revision: #1 $ --> <!--XML DTD for App Billing -->	DESCRIPTION	TXN DATA FORMAT	VALID FOR	NOTES
<!ELEMENT BILLING (CARRIER,DID,GMT,BR*, CT)>				
<!ELEMENT CARRIER (#PCDATA)>	<!-- carrier name -->	30an	all	
<!ELEMENT DID (#PCDATA)>	<!-- unique document ID -->	22an	all	DID is a unique document ID that identifies a specific carrier billing extract file. The DID is unique across all carriers.
<!ELEMENT GMT (#PCDATA)>	<!-- GMT time format MMDDYYYY HHMISS -->	MMDDYYYY HHMISS	all	
<!ELEMENT BR (EI,CID,ADS,SID,NSID,PN,NM,MF,SN,RF,LTS,GMT,TY,PM,PB,PV,PR,CY,DP,AR,AD,AP,AE,AI)>	<!-- billing record -->			
<!ELEMENT EI (#PCDATA)>	<!-- event id -->	12n	all	Numeric value, 12n maximum, not padded with leading zeros
<!ELEMENT CID (#PCDATA)>	<!-- sub carrier id -->	10n	all	Numeric value, 10n maximum, not padded with leading zeros
<!ELEMENT ADS (#PCDATA)>	<!-- download server -->	10n	SS, SB, SE, DL, DE	Numeric value, 10n maximum, not padded with leading zeros
<!ELEMENT SID (#PCDATA)>	<!-- subscriber id (phone number) -->	200an	SS, SB, SE,DL, DE, MI	note format of SID is determined by what the carrier provisions on the phone.
<!ELEMENT NSID (#PCDATA)>	<!-- new subscriber id (phone number) -->	200an	MI*	note format of new SID is determined by what the carrier provisions on the phone.
<!ELEMENT PN (#PCDATA)>	<!-- part number -->	30an	SS, SB, SE,DL,	sequential from QC1000 onwards for QC part numbers. QC part numbers will fill up to 10an.

			DE, MA*	External (i.e., carrier) provided part numbers may defined to up to 30an.
<!ELEMENT NM (#PCDATA)>	<!-- part name -->	30an	SE,DL, DE, MA*	
<!ELEMENT MF (#PCDATA)>	<!-- manufacturer -->	30an	SE,DL, DE, MA*	
<!ELEMENT SN (#PCDATA)>	<!-- subscription plan name -->	50an	SS, SB, SE	
<!ELEMENT RF (#PCDATA)>	<!-- restricted flag -->	1a	SS, SB, SE,DL, DE, MA	Y or N
<!ELEMENT LTS (#PCDATA)>	<!-- local time stamp -->	MMDDYYYY HHMISS	all	
<!ELEMENT TY (#PCDATA)>	<!-- Type -->	2an	all	SS = subscription start SB = subscription bill SE = subscription end DL = download – includes purchased, demo, upgrade, subscription, pre-install, trial DE = delete (permanent delete only, does not include disabled applications) TA = transaction adjustment MA = developer adjustment MI = MIN update (transfer or deactivation)
<!ELEMENT PM (#PCDATA)>	<!-- price method -->	2an	DL, SB	1 = demo value 2 = purchase value 3 = subscription value 4 = upgrade value 5 = preinstall value 6 = trial value
<!ELEMENT PB (#PCDATA)>	<!-- price basis -->	2an	DL, SB	1 = Number of uses value 2 = Fixed Expiration date value 3 = Fixed Duration value 4 = Elapsed Time value

<!ELEMENT PV (#PCDATA)>	<!-- price value -->	30an	DL, SB	Number of Uses: price value = count or unlimited = "unlimited" Fixed Expiration: price value = date expressed in GMT or unlimited = "unlimited" Fixed Duration: price value = number of days or unlimited = "unlimited" Elapsed Time: price value = number of minutes, 0= unlimited
<!ELEMENT PR (#PCDATA)>	<!-- price -->	S10.2n	DL, SB	sign digit (explicitly +/-), 10 characters, decimal point 2 digits total 14. Is in the currency which is used for list price in the carrier catalog
<!ELEMENT CY (#PCDATA)>	<!-- currency -->	2an	DL, SB	1 = U.S. 2 = Canadian 3 = Yen 5 = Peso 6 = Won
<!ELEMENT DP (#PCDATA)>	<!-- developer price -->	S10.2n	DL, SB	sign digit (explicitly +/-), 10 characters, decimal point 2 digits total 14. Is in U.S. currency
<!ELEMENT AR (#PCDATA)>	<!-- adjustment reason -->	2an	TA, MA	Transaction Adjustments 1 = Duplicate Download 2 = Accidental Download 3 = Customer Dissatisfaction 4 = Application Recall Developer Adjustments 5 = Consumer Bad Debt 6 = Part Number Adjustment 7 = General Adjustment
<!ELEMENT AD (#PCDATA)>	<!-- adjustment dap -->	S10.2n	TA, MA	sign digit (explicitly +/-), 10 characters, decimal point 2 digits total 14. Must be in U.S. currency

<!ELEMENT AP (#PCDATA)>	<!-- adjustment price -->	S10.2n	TA	sign digit (explicitly +/-), 10 characters, decimal point 2 digits total 14. Must be in the same currency as list price is displayed in catalog
<!ELEMENT AE (#PCDATA)>	<!-- adjustment description -->	240an	TA, MA	Free text description field
<!ELEMENT AI (#PCDATA)>	<!-- adjusted event id -->	12n	TA	Same format as event id, this identifies the original billing event to apply the adjustment to.
<!ELEMENT CT (#PCDATA)>	<!—record count -->	22n	All	Record count of BR records

Example Carrier Billing Extract XML File

```
<?xml version="1.0" ?>
- <BILLING>
    <CARRIER>Qualcomm</CARRIER>
    <DID>27</DID>
    <GMT>06172001 115347</GMT>
  - <BR>
      <EI>2957957</EI>
      <CID>101</CID>
      <ADS>brew-ads1</ADS>
      <SID>000008584495526</SID>
      <NSID>null</NSID>
      <PN>QC450</PN>
      <NM>Brick Attack</NM>
      <MF>Qualcomm</MF>
      <SN>null</SN>
      <RF>N</RF>
      <LTS>01012001 102246</LTS>
      <GMT>01012001 182246</GMT>
      <TY>DL</TY>
      <PM>2</PM>
      <PB>2</PB>
      <PV>30-jun-01</PV>
      <PR>+1.00</PR>
      <CY>1</CY>
      <DP>+1.00</DP>
      <AR>null</AR>
      <AD>null</AD>
      <AP>null</AP>
      <AE>null</AE>
      <AI>null</AI>
    </BR>
  - <BR>
      <EI>2957952</EI>
      <CID>101</CID>
      <ADS>brew-ads1</ADS>
      <SID>000008584495379</SID>
      <NSID>null</NSID>
      <PN>QC460</PN>
      <NM>Scheduler</NM>
      <MF>Qualcomm</MF>
      <SN>null</SN>
      <RF>N</RF>
      <LTS>06062001 135726</LTS>
      <GMT>06062001 215726</GMT>
      <TY>DL</TY>
```

```
47          <PM>3</PM>
48          <PB>3</PB>
49          <PV>30</PV>
50          <PR>+2.00</PR>
51          <CY>1</CY>
52          <DP>+2.00</DP>
53          <AR>null</AR>
54          <AD>null</AD>
55          <AP>null</AP>
56          <AE>null</AE>
57          <AI>null</AI>
58        </BR>
59      - <BR>
60          <EI>2957954</EI>
61          <CID>101</CID>
62          <ADS>brew-ads1</ADS>
63          <SID>000006198905831</SID>
64          <NSID>null</NSID>
65          <PN>QC461</PN>
66          <NM>Alarm Clock</NM>
67          <MF>Qualcomm</MF>
68          <SN>null</SN>
69          <RF>N</RF>
70          <LTS>06062001 140830</LTS>
71          <GMT>06062001 220830</GMT>
72          <TY>DL</TY>
73          <PM>1</PM>
74          <PB>4</PB>
75          <PV>15</PV>
76          <PR>+3.00</PR>
77          <CY>1</CY>
78          <DP>+3.00</DP>
79          <AR>null</AR>
80          <AD>null</AD>
81          <AP>null</AP>
82          <AE>null</AE>
83          <AI>null</AI>
84        </BR>
85      - <BR>
86          <EI>2957958</EI>
87          <CID>101</CID>
88          <ADS>brew-ads1</ADS>
89          <SID>000006198905831</SID>
90          <NSID>null</NSID>
91          <PN>QC462</PN>
92          <NM>Text Memos</NM>
93          <MF>Qualcomm</MF>
94          <SN>null</SN>
```

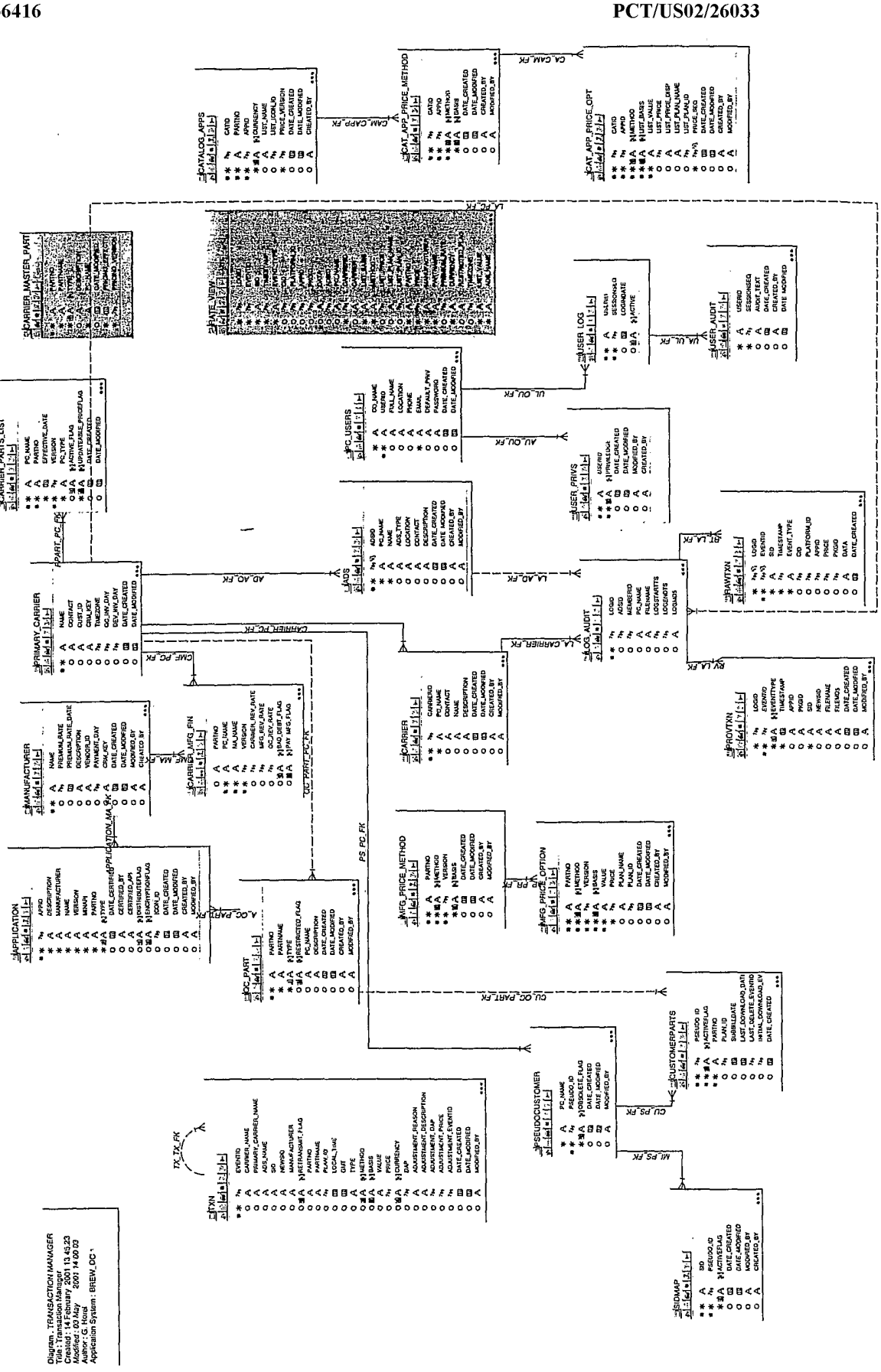
```
95          <RF>N</RF>
96          <LTS>06062001 150252</LTS>
97          <GMT>06062001 230252</GMT>
98          <TY>DL</TY>
99          <PM>2</PM>
100         <PB>3</PB>
101         <PV>30</PV>
102         <PR>+6.00</PR>
103         <CY>1</CY>
104         <DP>+6.00</DP>
105         <AR>null</AR>
106         <AD>null</AD>
107         <AP>null</AP>
108         <AE>null</AE>
109         <AI>null</AI>
110       </BR>
111     - <BR>
112         <EI>2957965</EI>
113         <CID>101</CID>
114         <ADS>brew-ads1</ADS>
115         <SID>000008584495545</SID>
116         <NSID>null</NSID>
117         <PN>QC461</PN>
118         <NM>Alarm Clock</NM>
119         <MF>Qualcomm</MF>
120         <SN>null</SN>
121         <RF>N</RF>
122         <LTS>06072001 075839</LTS>
123         <GMT>06072001 155839</GMT>
124         <TY>DL</TY>
125         <PM>1</PM>
126         <PB>4</PB>
127         <PV>15</PV>
128         <PR>+3.00</PR>
129         <CY>1</CY>
130         <DP>+3.00</DP>
131         <AR>null</AR>
132         <AD>null</AD>
133         <AP>null</AP>
134         <AE>null</AE>
135         <AI>null</AI>
136       </BR>
137     - <BR>
138         <EI>2957969</EI>
139         <CID>101</CID>
140         <ADS>brew-ads1</ADS>
141         <SID>000008584495545</SID>
142         <NSID>null</NSID>
```

```
143         <PN>QC461</PN>
144         <NM>Alarm Clock</NM>
145         <MF>Qualcomm</MF>
146         <SN>null</SN>
147         <RF>N</RF>
148         <LTS>06072001 131624</LTS>
149         <GMT>06072001 211624</GMT>
150         <TY>DL</TY>
151         <PM>1</PM>
152         <PB>4</PB>
153         <PV>15</PV>
154         <PR>+3.00</PR>
155         <CY>1</CY>
156         <DP>+3.00</DP>
157         <AR>null</AR>
158         <AD>null</AD>
159         <AP>null</AP>
160         <AE>null</AE>
161         <AI>null</AI>
162       </BR>
163     - <BR>
164         <EI>2957974</EI>
165         <CID>101</CID>
166         <ADS>brew-ads1</ADS>
167         <SID>000008584495545</SID>
168         <NSID>null</NSID>
169         <PN>QC462</PN>
170         <NM>Text Memos</NM>
171         <MF>Qualcomm</MF>
172         <SN>null</SN>
173         <RF>N</RF>
174         <LTS>06072001 132103</LTS>
175         <GMT>06072001 212103</GMT>
176         <TY>DL</TY>
177         <PM>2</PM>
178         <PB>3</PB>
179         <PV>30</PV>
180         <PR>+6.00</PR>
181         <CY>1</CY>
182         <DP>+6.00</DP>
183         <AR>null</AR>
184         <AD>null</AD>
185         <AP>null</AP>
186         <AE>null</AE>
187         <AI>null</AI>
188       </BR>
189     - <BR>
190         <EI>2958003</EI>
```

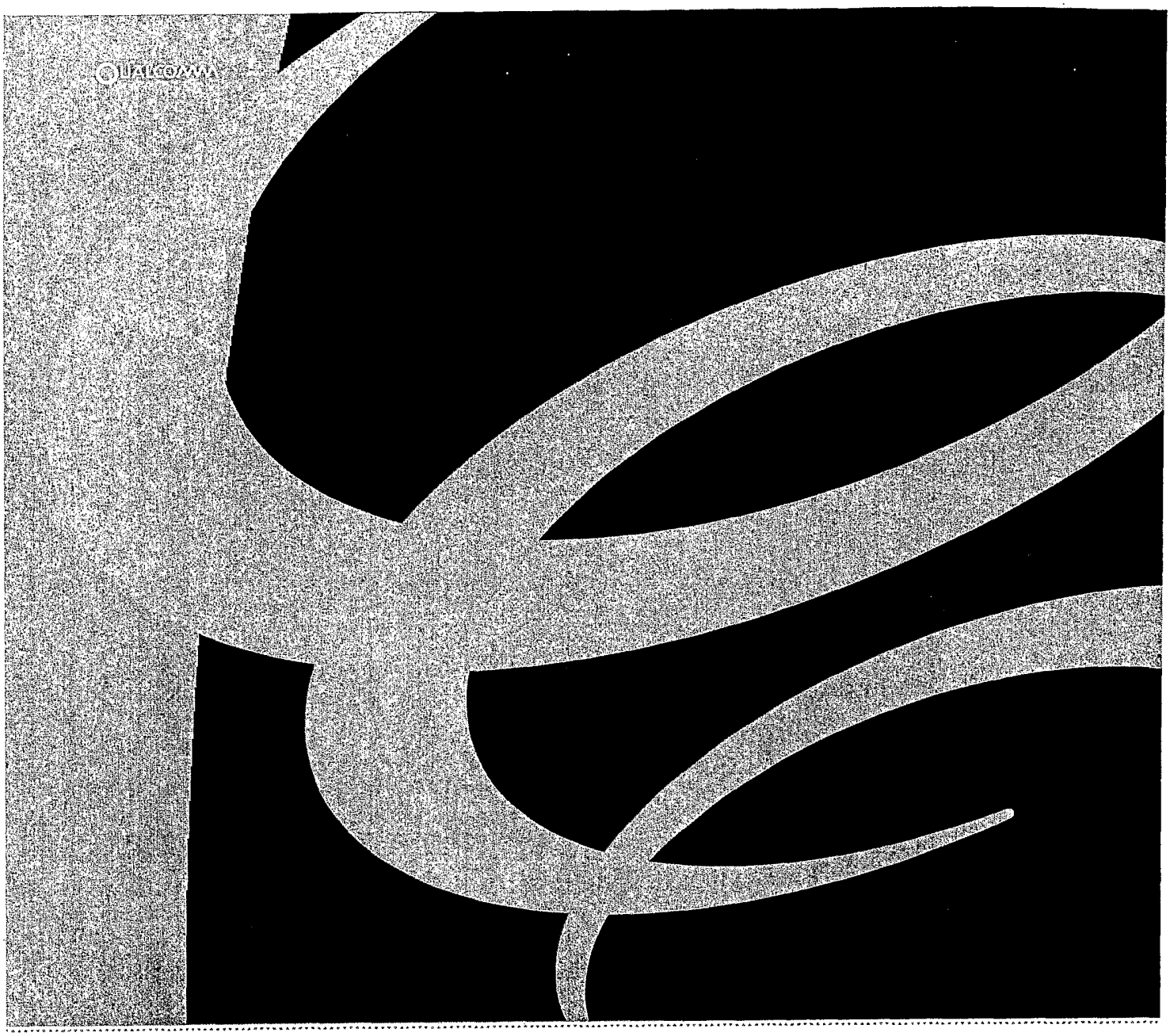
```
191         <CID>101</CID>
192         <ADS>brew-ads1</ADS>
193         <SID>000008584495545</SID>
194         <NSID>null</NSID>
195         <PN>QC450</PN>
196         <NM>Brick Attack</NM>
197         <MF>Qualcomm</MF>
198         <SN>null</SN>
199         <RF>N</RF>
200         <LTS>06112001 070604</LTS>
201         <GMT>06112001 150604</GMT>
202         <TY>DL</TY>
203         <PM>1</PM>
204         <PB>2</PB>
205         <PV>27-mar-01</PV>
206         <PR>+0.00</PR>
207         <CY>1</CY>
208         <DP>+0.00</DP>
209         <AR>null</AR>
210         <AD>null</AD>
211         <AP>null</AP>
212         <AE>null</AE>
213         <AI>null</AI>
214       </BR>
215       <CT>8</CT>
216     </BILLING>
217
```

218 # Appendix A – Transaction Manager Data Model

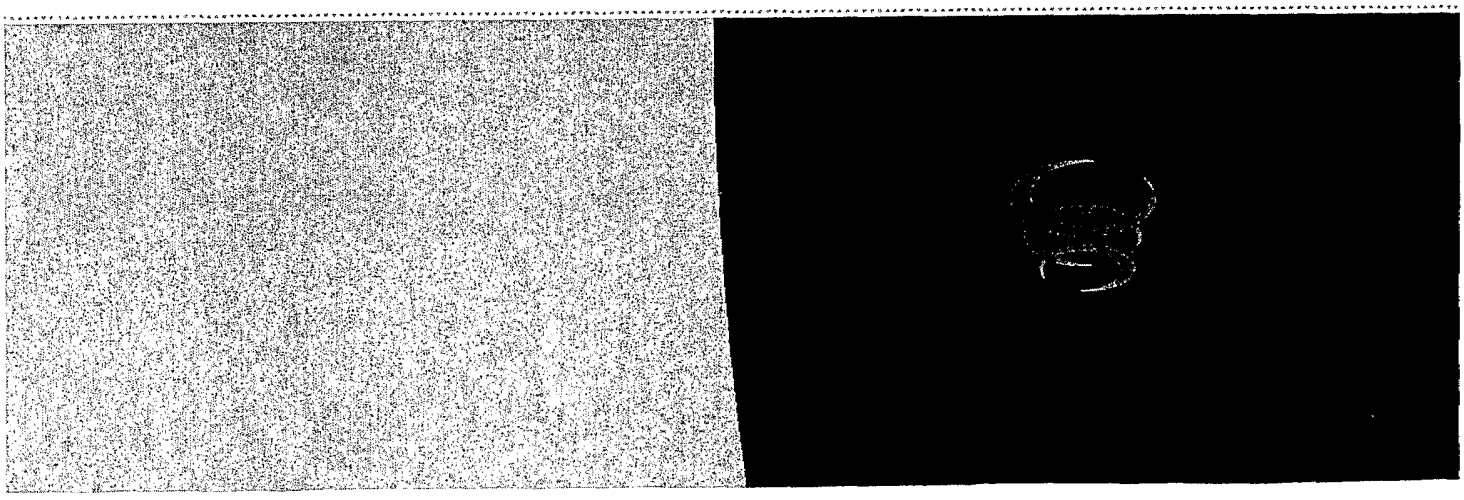
219 This section defines the Transaction Manager Data Model for the TX schema


Transaction Manager High Level Overview
Diagram : TRANSACTION MANAGER
Title : Transaction Manager High Level Overview
Created : 14 February 2001 13:45:23
Modified : 03 May 2001 14:00:03
Author : G. Hood
Application System : BREW_DC
CARRIER_MASTER_PART
CARRIER_PARTS_LIST
PRIMARY_CARRIER
MANUFACTURER
APPLICATION
CARRIER_MFG_PIN
DC_PART
MFG_PRICE_METHOD
MFG_PRICE_OPTION
CARRIER
PC_USERS
USER_LOG
USER_AUDIT
USER_PRIVS
LOG_AUDIT
AIOS
CATALOG_APPS
CAT_APP_PRICE_METHOD
CAT_APP_PRICE_OPT
RAWTXN
PROVTXN
TXN
PSEUDOCUSTOMER
CUSTOMERPARTS
SIDMAP

VALUE	TEXT	TYPE
Y	YES	YESNO
N	NO	YESNO
1	US Dollars	CURRENCY
6	Won	CURRENCY
3	Yen	CURRENCY
2	Canadian Dollars	CURRENCY
5	Peso	CURRENCY
DE	Delete	EVENT
PR	Pre-Install	EVENT
UA	Unactivate	EVENT
SS	Subscription Start	EVENT
TR	Transfer	EVENT
SB	Subscription Bill	EVENT
TA	Transaction Adjustment	EVENT
DA	Developer Adjustment	EVENT
SE	Subscription End	EVENT
DL	Download	EVENT
3	Subscription	PRICE_METHOD
4	Upgrade	PRICE_METHOD
1	Demo	PRICE_METHOD
5	PreInstall	PRICE_METHOD
2	Purchase	PRICE_METHOD
1	Uses	PRICE_BASIS
4	Elapsed Time	PRICE_BASIS
2	Expiration Date	PRICE_BASIS
3	Days	PRICE_BASIS



Binary Runtime Environment for Wireless™ | Carrier's Guide to Application Distribution



brew

Contents

brew

Introduction

This guide contains information about and instructions for distributing BREW applications to device users. From a carrier's point of view, four phases comprise application distribution:

- Populating the Carrier Parts List
- Creating and managing catalogs
- Administrating the Application Download Servers (ADS)
- Monitoring application usage

Virtually all phases of application distribution are handled via the BREW Carrier Extranet. Detailed instructions for each phase appear in the remaining sections of this document, as shown below.

Section	Description
Basic Concepts	Provides an overview of the application distribution process, discusses terminology, and gives instructions for navigating the BREW Carrier Extranet.
Populating the Carrier Parts List	Introduces the Master Parts List, and gives instructions for adding applications to the Carrier Parts List.
Creating and Managing Catalogs	Provides instructions for creating and managing catalogs and the categories and applications they contain.
Distributing the Applications	Talks about BREW Application Download Servers and tells you how to move catalogs to selected servers.
Understanding Reports	Discusses BREW billing adjustments, queries, and reports.

Related documents

This document is part of an information set. Other BREW documents include:

Document	Description
Developer's Guide to BREW Application Distribution	Explains the BREW distribution process and provides instructions for submitting applications for certification. Also discusses the developer/carrier relationship.
BREW SDK User's Guide	Introduces the components of the BREW Software Development Kit (SDK) and their relationships to one another. The document also contains general instructions for developing BREW applications. Auxiliary documents include the *BREW Device Configurator Guide, BREW Resource Editor Guide,* and *BREW MIF Editor Guide.*
BREW API Reference	Provides programmers with information about BREW functions and data structures needed to develop applications for BREW-enabled mobile platforms.

For more information

Online information and support is available for carriers. Please visit the BREW web site for details: *www.qualcomm.com/brew.*

brew

Basic Concepts

This section introduces the basic concepts of BREW application distribution and describes the BREW Carrier Extranet.

Before you begin

While a lot of thought and effort has gone into making the interface as user-friendly as possible, you will probably find the process more intuitive if you first familiarize yourself with the following terminology:

Application	An application that has been developed on the BREW platform and has been properly tested and certified.
Application Download Server (ADS)	The server that handles the transfer of applications selected for purchase by device users. Devices are provisioned to connect to a particular ADS, and the catalog that resides on that particular ADS contains the applications that will display on a user's device.
Catalog	A database structure that houses applications assigned to categories. You can create and edit multiple catalogs targeted to various groups of users. When a user accesses MobileShop on a device, he or she sees the applications contained in one catalog.
Category	A grouping of related applications within a catalog. For example, you might want to create one category called "Games," another called "Business Applications," and so on. An application can be listed in multiple categories within a single catalog.
Carrier Parts List	The list of all applications that you have chosen from the Master Parts List. When you add applications to catalogs, you will choose the applications from this list.
Independent Software Vendor (ISV)	The party from whom the application is purchased.
Master Parts List	The list of all applications available to a certain carrier. When you choose an application from this list, it is added to the Carrier Parts List, from where you can later add it to a catalog.
MobileShop	The software utility through which users purchase and download applications from the ADS to their devices.

Security roles

While the Extranet provides access to all phases of BREW application distribution, certain pages may not be visible to certain users, depending on their assigned security roles:

Security Role	Accessible Pages (and related subpages)
Carrier Administrator	All pages except billing information
Catalog Administrator	Build/Edit Catalog
ADS Administrator	Assign Catalog to ADS
Billing/Finance	View Reports and Status

About the Carrier Extranet

The BREW Carrier Extranet provides the interface to all phases of application distribution. A brief description of each page of the Extranet follows.

Home page

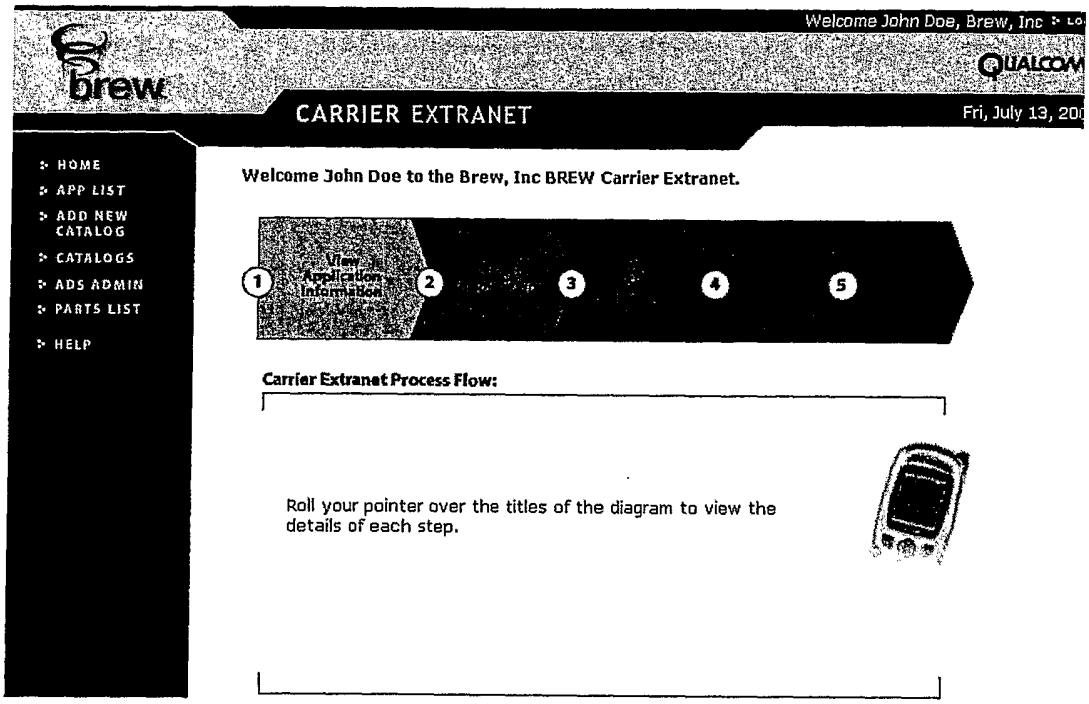
To open the Carrier Extranet Home page

1 Open your Internet browser and enter ***http://brewx.qualcomm.com/carrier/.*** Then log in with your user ID and password.

2 Enter your user name and password, and press **Enter**.

*The **Home** page appears.*

INDEX →



Move the mouse pointer over the scrolling bar to see a brief functional description of each part of the Extranet. The left side of the page contains an index of callable pages. Brief descriptions of each main Extranet page follow.

View Application Information

This page contains the **Carrier Parts List,** which is a listing of applications you have chosen from the Master Parts List.

Game 1
Game 2

QUALCOMM Parts List

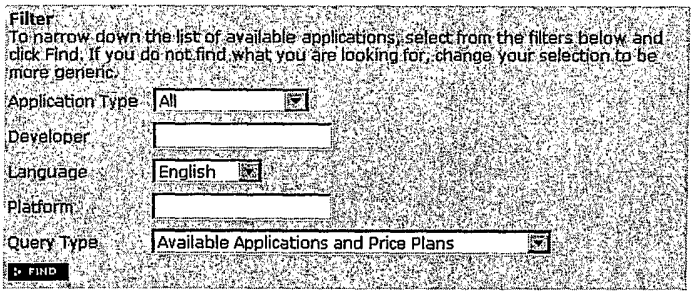
Part Name	Type	Restricted Flag	Description
[redacted]	G	N	[redacted] signed application
[redacted]	G	N	[redacted] signed application
Scheduler	P	N	*a signed sheduling application*
Alarm Clock	U	N	Alarm clock signed application
Text Memos	U	N	Text Memos signed application
Calculator	U	N	Calculator signed application
RoadWarrior	S	N	Real-time traffic speeds signed application
Sokoban	G	N	Sokoban signed application
Blackjack	G	N	Blackjack signed application
Go	O	N	Go signed application
Stock Master	S	N	Stock Master signed application
Hangman	G	N	Hangman signed application
Solitaire	G	N	Solitaire signed application

Clicking on a Part Name calls a sub-page of detailed information for the selected part, as discussed in the next section (Populating the Carrier Parts List).

Select Applications

This page contains the **Master Parts List**, which contains all new BREW applications available to your organization. In this context, "new" refers to those applications which you have not yet added to the Carrier Parts List, as well as upgrades to applications that are already in your Carrier Parts List. You can specify filtering criteria that will narrow the list to encompass only those applications of interest.

APPLICATIONS LIST

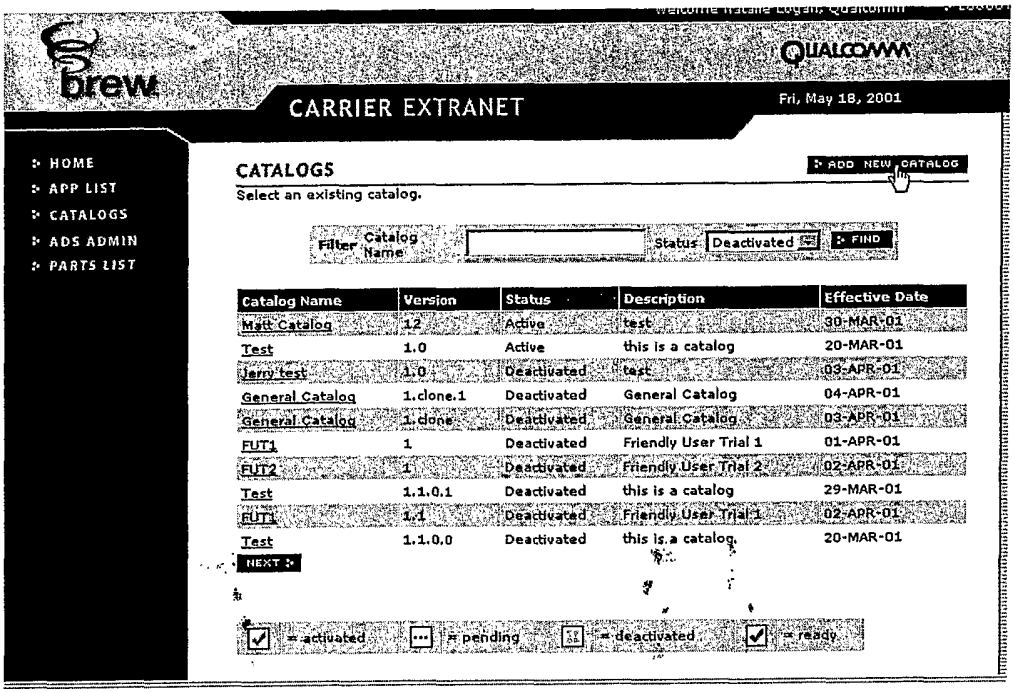


Application Name	Price Plan	Type	Description	Developer
[redacted] Game1	2	Games	[redacted] Game1	Qualcomm
[redacted] Game1	2	Games	[redacted] Game1	Qualcomm
[redacted] Service1	0	Services	[redacted] Service1	Qualcomm
[redacted] Service1	0	Services	[redacted] Service1	Qualcomm
Scheduler	1	Productivity	Scheduler	Qualcomm
Scheduler	1	Productivity	Scheduler	Qualcomm
Sokoban	0	Games	Sokoban	Qualcomm
Sokoban	0	Games	Sokoban	Qualcomm

When no new applications are available, the `No new apps` message appears. For more information, see the section entitled Populating the Carrier Parts List.

Build/Edit Catalogs

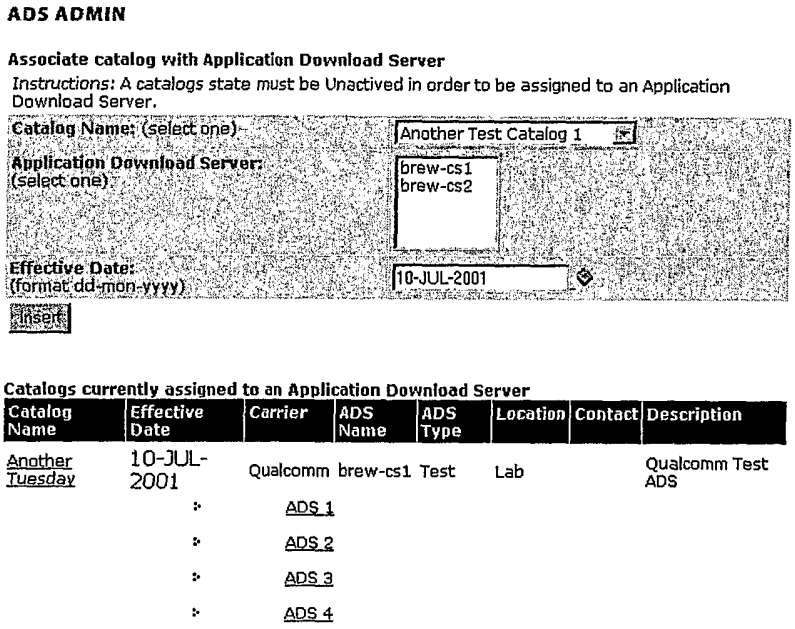
This page lists your existing catalogs, and lets you create new catalogs. Clicking on the name of an existing catalog calls pages that allow you to edit the catalog, create catalog categories, and add applications to the catalog.



For more information, see the section entitled Creating and Managing Catalogs.

Assign Catalog to ADS

This page is where you assign catalogs to specific Application Download Servers, from where specified applications can be downloaded by device users.



For more information, see the section entitled Distributing the Applications.

View Reports and Status

This page lets you choose from a variety of carrier-specific reports that will assist you in understanding your statement from QUALCOMM and in billing your customers.

HELP

The HELP page provides online support for carriers, and instructions for contacting the technical support center.

brew

Populating the Carrier Parts List

This section contains instructions for choosing the applications that you will later place in catalogs.

Before you can build catalogs of BREW applications, you must first populate the **Carrier Parts List** by choosing from applications in the **Master Parts List**. The Master Parts List contains new applications that are available to your organization, as well as existing applications for which the ISV has updated the pricing.

You can streamline the application selection process by setting specific search criteria. Suppose, for example, that you are interested in new games, but not in other application types. You can simply set that preference as one of your filtering criteria, and only new games will be displayed in the Master Parts List.

Choosing Applications

The following procedure guides through the process of choosing applications from the **Master Parts List** and adding them to your **Carrier Parts List**. This is accomplished in three steps:

1 Setting the filtering criteria

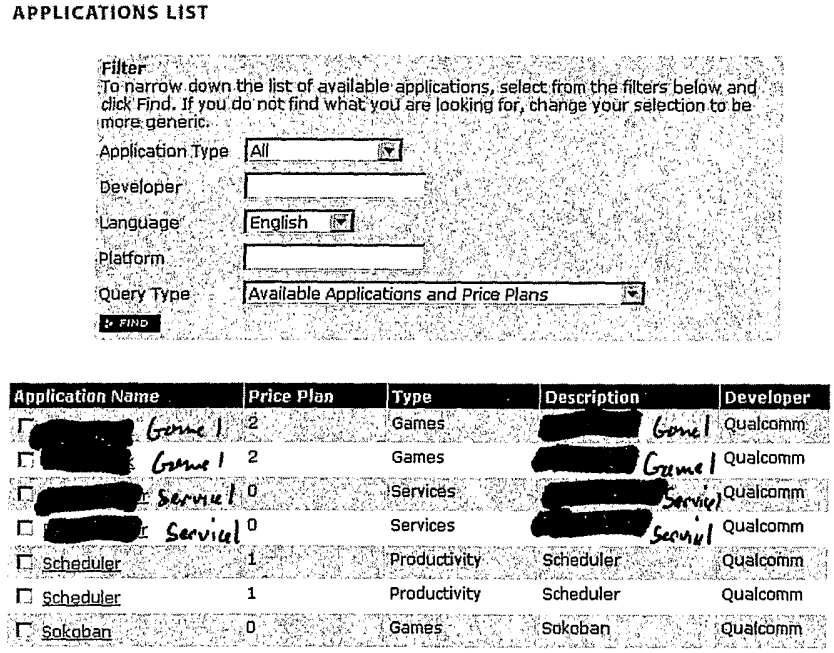
2 Choosing the application

3 Approving the price plan

To choose applications from the Master Parts List

1 From the **Home** page, choose **Select Applications.**

*The **Applications List** page appears. The Filter dialog box occupies the top part of the page, and the Master Parts List occupies the lower portion.*



2 If desired, use the **Filtering** dialog box to enter criteria to restrict the search.

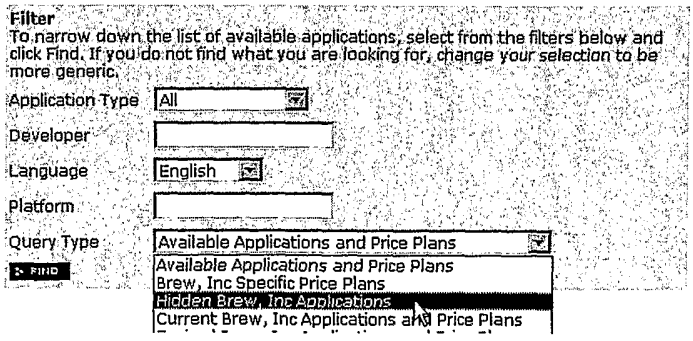
a For **Application Type**, select the application type (for example, **Game**) from the list.

b For **Developer**, enter the name of the ISV.

c For **Language**, select the name of the language in which the application displays on a device.

d For **Platform**, enter the device platform.

e For **Query Type**, select the scope of the applications to be searched:



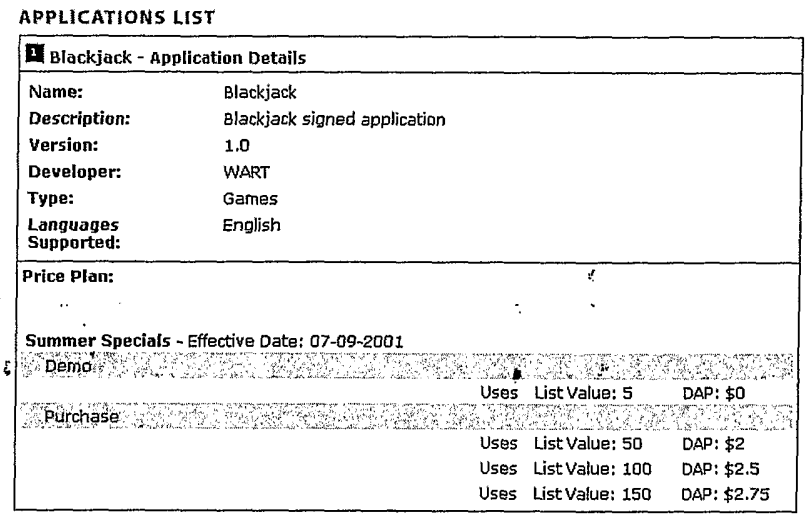
f To suppress the display of any application, click the checkbox to the left of its name and then click **Hide Checked**. To suppress the display of all applications, click **Hide All**. To redisplay a hidden application, select **Hidden applications** from the Query Type list.

3 When you have finished, click **Find**.

Only those applications that match your criteria will appear in the Master Parts List.

4 From the **Master Parts List** at the bottom of the page, choose the application you want to add to the **Carrier Parts List** by clicking on the application name.

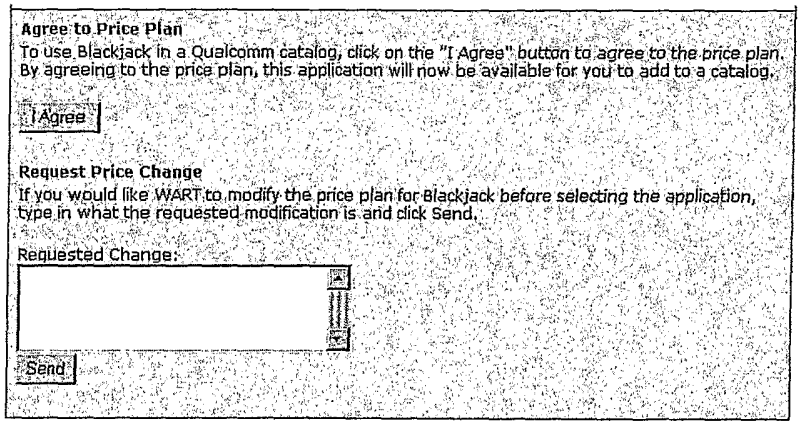
The Application Details page appears. The top portion of the page (shown below) contains details about the application and the price plan submitted by the ISV.



Explanations of the Application Details follow.

Name	The name of the application.
Description	An optional application description supplied by the software supplier.
Developer	The name of the ISV who originated the application.
Version	The software version number
Type	The type of the application. BREW applications must be one of the following types: Communications, Games, Internet, Music, Other, Productivity, Service, or Utility.
Languages Supported	The languages supported by the application.

The bottom portion of the page (shown below) contains a dialog box wherein you can either agree to the price plan ***or*** *request a change to the price plan..*



a To accept the price plan as submitted, click **I Agree.**

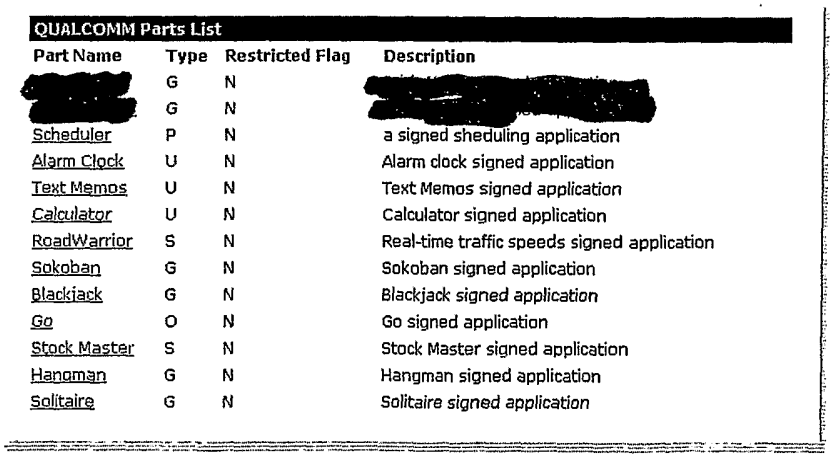
The application is added to your ***Carrier Parts List.***

b To request a change to the price plan, type a message to the ISV in the text window provided. When you are finished, click **Send.**

NOTE: *Your message is sent to QUALCOMM and subsequently to the ISV, who will contact you.* ***The application is not added to your Carrier Parts List until you have agreed to a price plan.***

About the Carrier Parts List

The **Carrier Parts List** shows the applications that you have previously selected from the **Master Parts List.**



QUALCOMM Parts List

Part Name	Type	Restricted Flag	Description
[redacted]	G	N	[redacted]
[redacted]	G	N	
Scheduler	P	N	a signed sheduling application
Alarm Clock	U	N	Alarm clock signed application
Text Memos	U	N	Text Memos signed application
Calculator	U	N	Calculator signed application
RoadWarrior	S	N	Real-time traffic speeds signed application
Sokoban	G	N	Sokoban signed application
Blackjack	G	N	Blackjack signed application
Go	O	N	Go signed application
Stock Master	S	N	Stock Master signed application
Hangman	G	N	Hangman signed application
Solitaire	G	N	Solitaire signed application

Brief definitions of the list's four columns follow.

Part Name	The name of the application, in English.
Type	The type of the application. BREW applications must be one of the following types: Communications (C), Games (G), Internet (I), Music (M), Other (O), Productivity (P), Service (S), or Utility (U).
Restricted Flag	Indicates whether this is a standard application or a carrier-restricted application.
Description	An optional application description supplied by the software supplier.

brew

Creating and Managing Catalogs

Five basic tasks make up the catalog management portion of the BREW application distribution process:

- Creating catalogs
- Creating catalog categories
- Adding applications to categories
- Manipulating application pricing
- Making the catalog ready for use

The following subsections contain instructions for performing each of these tasks.

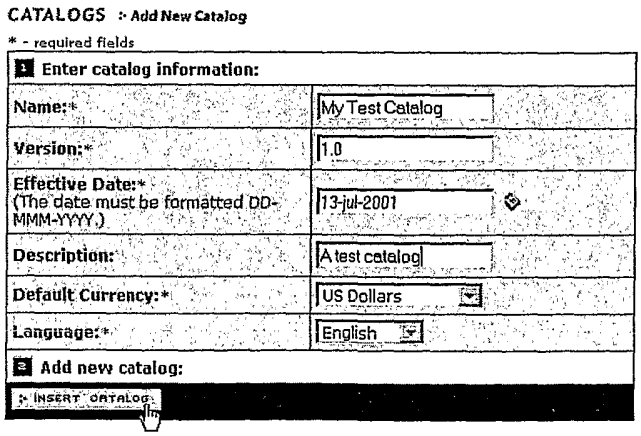
Creating a new catalog

Following is a step-by-step procedure for creating a new catalog.

To create a new catalog

1 From the index on the left side of the Home page, select **Add New Catalog.**

*The **Add New Catalog** page appears.*



a For **Name** (required), enter a name for your catalog (for example, My Test Catalog).

b For **Version** (required), enter the catalog version number. This is an internal alphanumeric identifier that can help you keep track of catalog iterations.

c For **Effective Date** (required), enter the date that the catalog is expected to go "live" -- that is, the date on which you plan to make its applications available for purchase by device users. Enter the date in the format shown, or click on the icon to the right of the **Effective Date** field to display a calendar.

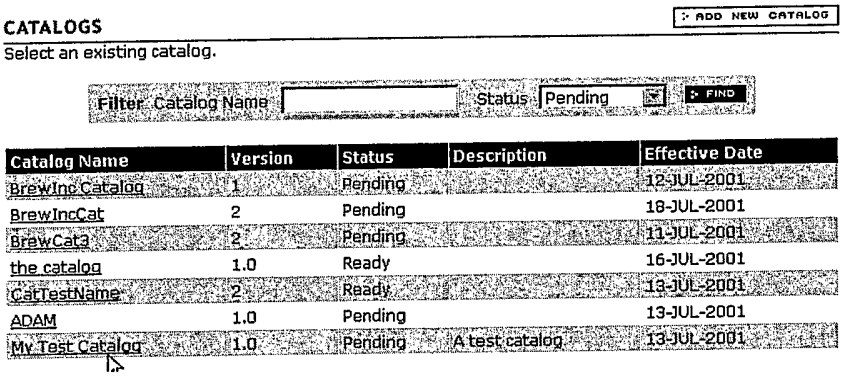
d For **Description** (optional), enter any other information that will help identify your catalog.

e For **Default Currency**, select the currency from the list. The **Default Currency** setting forces every application in this catalog to display in the currency specified. The symbol for the specified currency will display on the device, with the application purchase price.

f For **Language**, choose a language option from the list. (See About the language setting for more information.)

2 When you are finished, click **Insert Catalog**.

*The new catalog appears in the list on the **Catalogs** page.*



Notice that the status of the catalog has automatically been set to Pending. (See About catalog status for more information.)

About the language setting

The language setting for a catalog determines the language displayed on targeted devices. If you set the language to Korean, for example, the catalog categories display in Korean, and only Korean-language applications display on the device.

While a device might support multiple languages, it displays only one language at a time. Users of devices with multilingual capabilities can designate their language of choice, and the phone will display only those catalogs that have the designated language setting.

About catalog status

A catalog can be in one of four states: Pending, Ready, Active, and Deactive.

- **Pending** means that the catalog is still undergoing changes. This is the only state in which a catalog can be edited.
- **Ready** means that the application is ready to be moved to an ADS.
- **Active** means that the catalog is in use on an ADS.
- **Deactive** means that the catalog has been taken out of commission. Once deactivated, a catalog cannot be made active again.

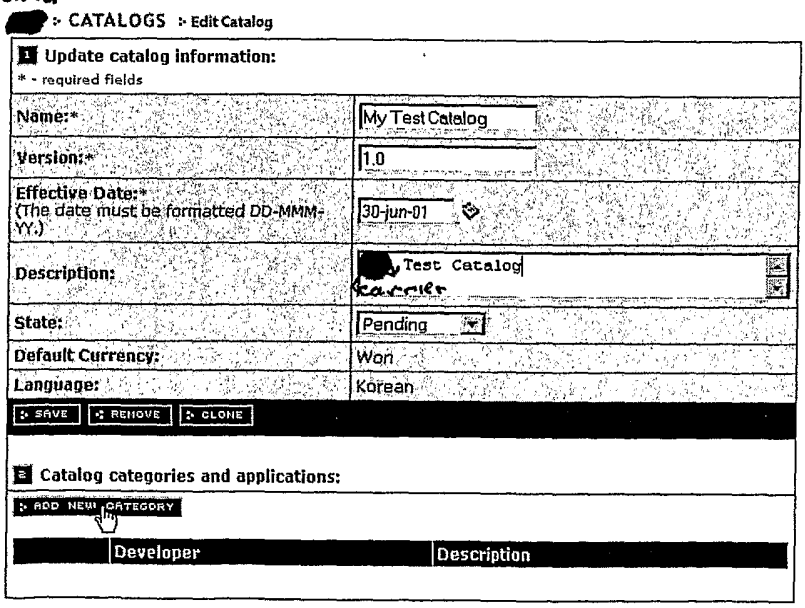
Creating categories

After you create a catalog, you can organize the applications it contains into logical categories. For example, you might create a category called "Games" and another called "Business Applications."

To create a category

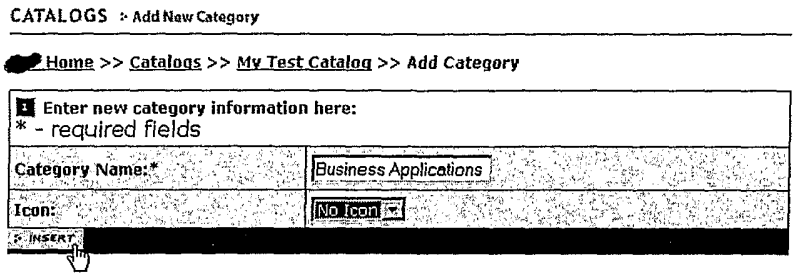
1 From the **Catalogs** page, click the name of your catalog.

*The **Edit Catalog** page appears.*



2 From the **Edit Catalog** page, click **Add New Category**.

*The **Add New Category** page appears.*

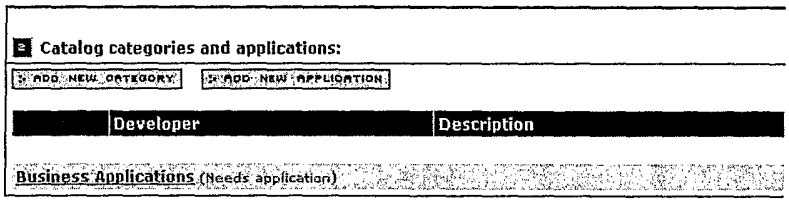


a For **Category Name**, enter a descriptive name for the category. The text string you enter is exactly what device users will see when they access MobileShop.

b From the **Icon** list, choose the icon number that corresponds to the icon you want to display for the category, or choose **No icon**.

3 When you have finished, click **Insert** to add the category to the catalog.

*The new category appears at the bottom of the **Edit Catalog** page.*



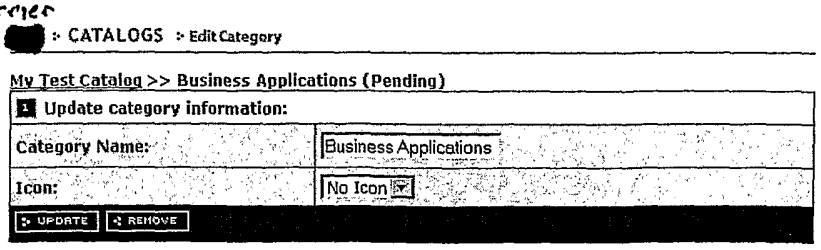
Editing or removing a category

You can edit the name or icon of a category after it is created, or you can remove a category from a catalog.

To edit a category

1 From the Edit Catalog page, click on the name of the category (for example, Business Applications).

The Edit Category page appears.



a To edit the category, enter your changes and then click **Update.**

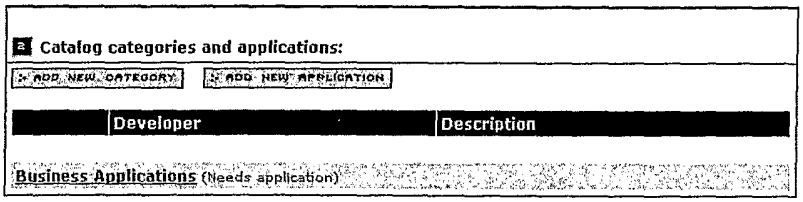
b To remove the category from the catalog, click **Remove.**

Adding applications

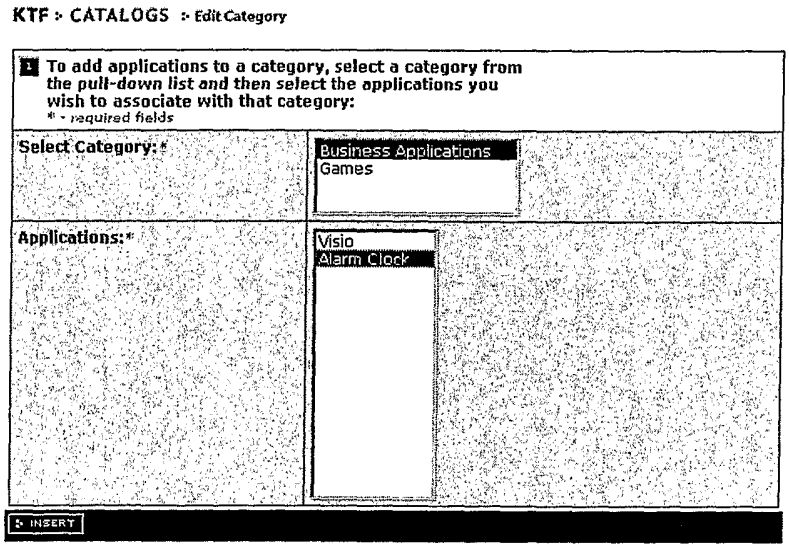
After you have created a category, you must select the applications that will belong to it.

To add applications to a category

1 From the **Edit Catalog** page, click **Add New Application.**



*The **Add Application** page appears.*

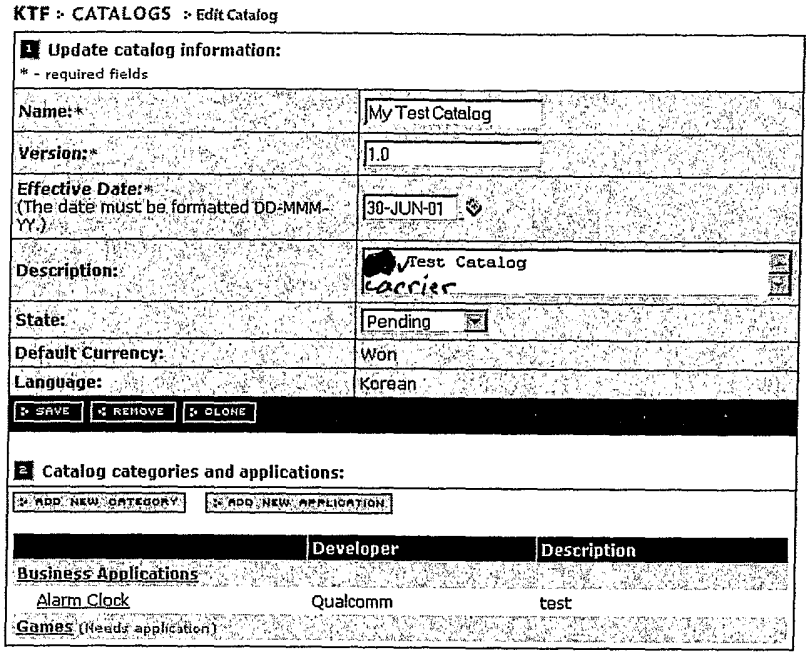


2 From the **Applications** list, select the applications that will belong to this category, in this catalog. Press and hold the **Ctrl** key to select multiple applications.

NOTE: Applications can belong to more than one category.

3 When you have finished selecting applications for the category, click **Insert.**

*Section 2 (bottom) of the **Edit Catalog** page now shows the applications contained in the category.*



Manipulating application price data

There are two prices associated with each application: the purchase price you charge device users, and the price charged by the software supplier. You can interactively manipulate the purchase price for each application; however, you cannot directly make changes to the price plan that was set by the supplier. If you wish to change the price plan for an application, you must negotiate with the software supplier and have the software supplier resubmit the price plan data.

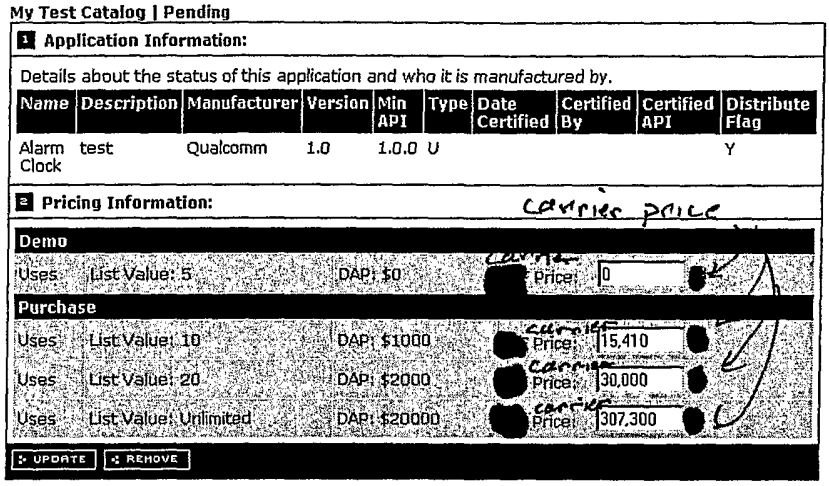
Specifying the user purchase price

You will need to set the purchase price of *each* application, in each catalog. Set the purchase price only once for each application in a single catalog, whether or not the application belongs to multiple categories. Its purchase price must be the same throughout the catalog.

To change the user purchase price for an application

1 From the **Edit Catalog** page, select an application from Section 2, Catalog categories and Applications.

*The **Edit Application** page appears.*



2 Adjust the prices, entering the amount that device users will pay to download the application. You must enter a price (including the currency symbol) for each pricing option on the page, even if there is no charge ($0.00). The Developer Application Price (DAP) -- the price charged by the software supplier -- is always given in U.S. currency, independent of the currency specified for the purchase price.

3 When you have completed the pricing information for that application, click **Update**.

NOTE: You must enter prices for each application in a catalog. If the application resides in multiple catalogs, you must set the price in each catalog.

Removing an application

You can remove an application from a category by clicking **Remove** on the **Edit Application** page (shown above).

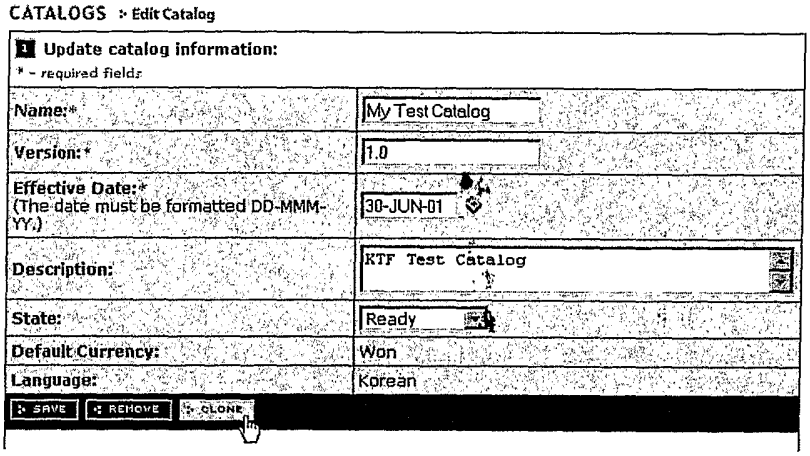
Cloning a catalog

If you wish to create a catalog that is similar to one already created, you might find it helpful to clone the existing catalog.

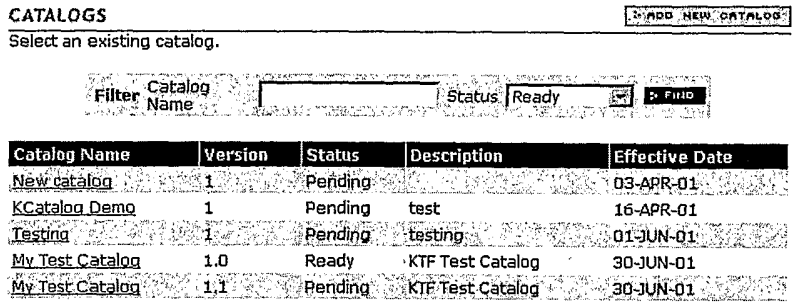
You can clone a catalog in any state, including active and deactive. Although active and deactive catalogs cannot be edited directly, their clones are editable because they are created in pending state.

To clone a catalog

1 From the **Edit Catalog** page, click **Clone.**



The Catalogs page appears, showing the new catalog. The version number of the new (clone) catalog has been revised automatically.

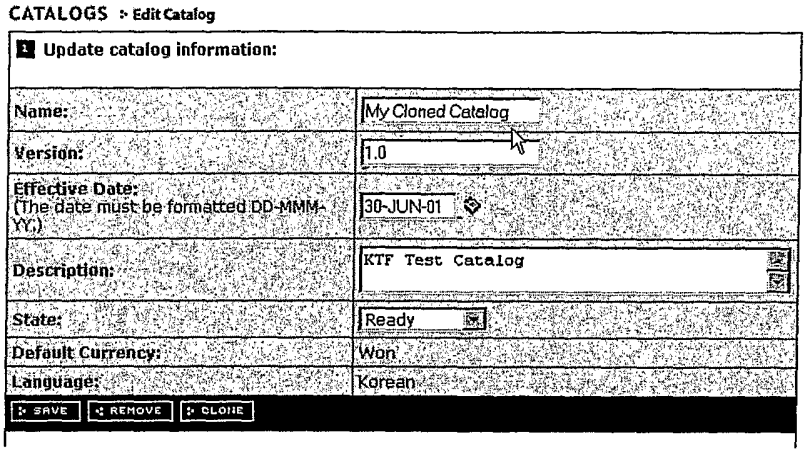


Catalog Name	Version	Status	Description	Effective Date
New catalog	1	Pending		03-APR-01
KCatalog Demo	1	Pending	test	16-APR-01
Testing	1	Pending	testing	01-JUN-01
My Test Catalog	1.0	Ready	KTF Test Catalog	30-JUN-01
My Test Catalog	1.1	Pending	KTF Test Catalog	30-JUN-01

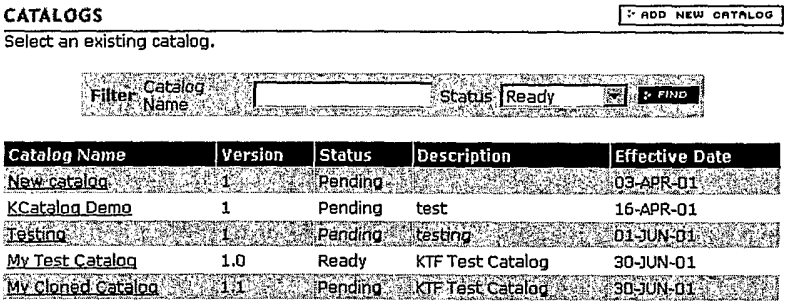
2 To edit the new catalog (for example, to change its name), click on the name.

The Edit Catalog page appears.

3 From the **Edit Catalog** page, select the information you want to change (for example, Name), and then click **Save.**



The new name appears in the list on the **Catalogs** page.



Finding catalogs quickly

The catalog management interface provides navigation tools that make it easy to find a particular catalog in your catalog list.

When you open the **Catalogs** page, this navigation bar appears at the top:

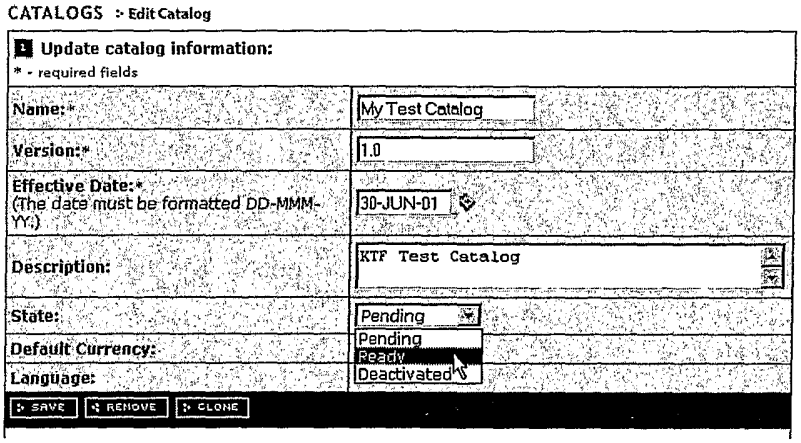
If you know the name of the catalog, or any part of the name, enter it in the **Catalog Name** field. If you know the state of the catalog, select it from the **Status** list. Then click **Find.** The only catalogs listed will be the ones that match your specifications.

Putting the catalog in Ready state

When you have finished all the catalog edits, including adjusting the prices for each application, your catalog is ready to be moved to an ADS. You must now change the status of the catalog from **Pending** to **Ready.**

To change the status of the catalog

1 From the **Edit Catalog** page, select **Ready** from the list in the **State** field.



The catalog is now ready to be moved to an ADS.

NOTE: Once the catalog is changed to Ready state, it is not editable. To make more changes to the catalog prior to activating it, use the **Edit Catalog** page to change the state back to **Pending**. Then make your changes, and change the status back to **Ready.**

See the section entitled Distributing the Applications for information about moving the catalog to an ADS.

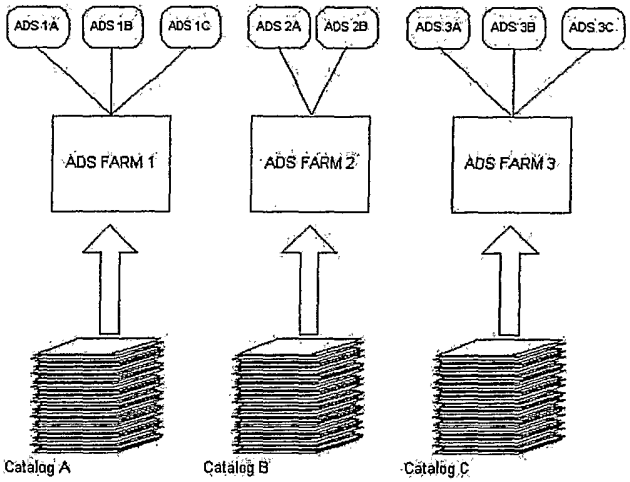
brew

Distributing the Applications

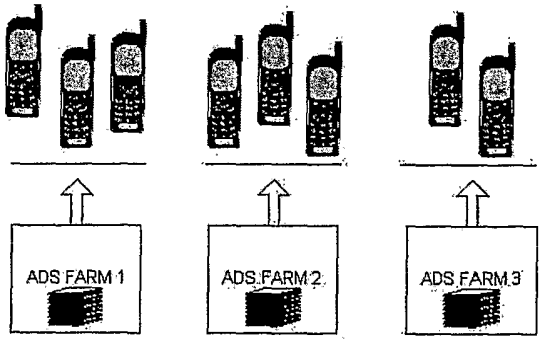
The actual distribution of BREW applications is accomplished via a flexible client/server architecture whereby the clients are the mobile devices requesting applications from the ADS. The ADS then allows the device to download the purchased applications to the devices provisioned for that ADS. Each ADS is assigned only one applications catalog. Therefore, if you have multiple catalogs, you will have multiple ADS's, each with its own clients (devices).

> **NOTE:** Although each ADS can accommodate only one catalog, the same catalog can be assigned to multiple ADS's.

In this document, all references to the ADS apply to a group of servers logically associated with an ADS "farm," as shown below.



First, catalogs in Ready state are assigned to ADS's. Only one catalog can be assigned to an ADS.



When you move a catalog to an ADS, it becomes visible to its provisioned devices (clients). Device users purchase and download applications through MobileShop.

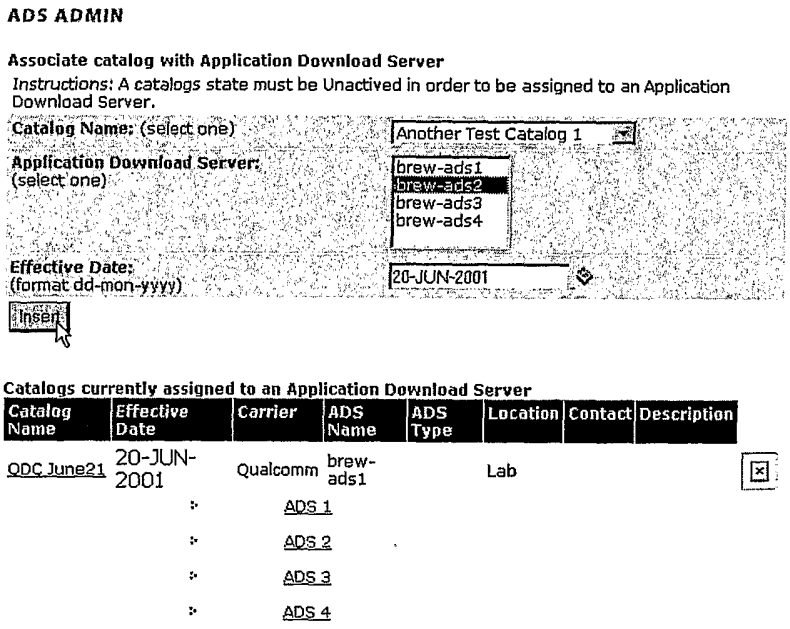
When a device is provisioned, it is given the IP address of the ADS containing the appropriate application catalog. For information about provisioning devices, refer to Name of Document.

After you have placed a catalog in Ready state, it is ready to be associated with an Application Download Server.

To assign a catalog to an ADS

1 From the index on the **Home** page, choose **ADS Admin.**

The ADS Admin page appears.



a For **Catalog Name,** choose the catalog to be activated.

b For **Application Download Server,** choose the ADS to which to propagate the catalog.

c For **Effective Date,** enter the date in the format specified or click on the icon to select a date from the calendar.

2 When you have finished, click **Submit.**

From the device user's point of view

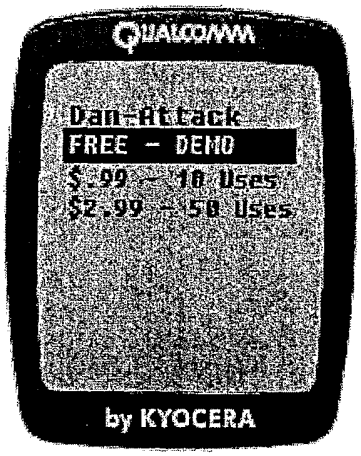
The following describes how a hypothetical device user can purchase an application via MobileShop.

1 User starts BREW from her mobile phone.

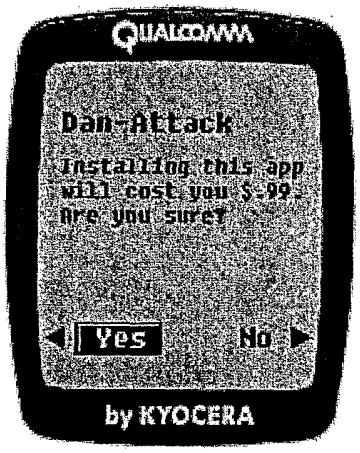
The ADS for which the device is provisioned displays the catalog of applications.

2 From the catalog that appears, the user chooses a category called **Games**. She notices new listing.

3 She clicks on the new application name, and is offered the opportunity view a quick demo of the game, along with some pricing information.



4 After viewing the demo, she decides to download the game. She selects the pricing option she wants and is prompted to confirm her choice.



1 Introduction

1.1 Purpose of This Document

The QDC Billing Technical Specification identifies and discusses the technical architecture for the BREW finance functions. These functions include the custom BREW billing event processing, PeopleSoft (AR, AP, and GL) functions, and extranet billing services (i.e., developer and carrier).

The purpose of the QDC Billing Technical Specification is the following:

- Serve as the primary document for technical design relating to BREW finance functions.
- Facilitate communication and agreement relating to functional and technical consistency across QDC teams and QIS management.
- Serve as an evolving document for specifying Phase 1 and future technical design enhancements.

1.2 Document Organization

This document contains the following sections:

- *Section 1, Introduction:* Describes the purpose of the document, document organizational structure, related documents and revision history.
- *Section 2, Technical Design:* Identifies the technical design by functional modules which include: carrier invoicing, developer payment, carrier extranet, developer extranet, and QIS billing tools.
- *Section 3, System Characteristics:* Describes the system characteristics of the BREW finance functions to include: server configuration, minimum downtime characteristics, degraded mode of operations, etc.
- *Appendix A, Glossary:* Defines acronyms, and abbreviations used in the document.
- *Appendix B, Invoices/Summaries/Vouchers:* Shows examples of the Developer fee invoice, the Qualcomm fee invoices, and the Developer payment vouchers.
- *Appendix C, Peoplesoft Configuration::* Describes how PeopleSoft is set up to support QIS Billing.

2 Technical Design

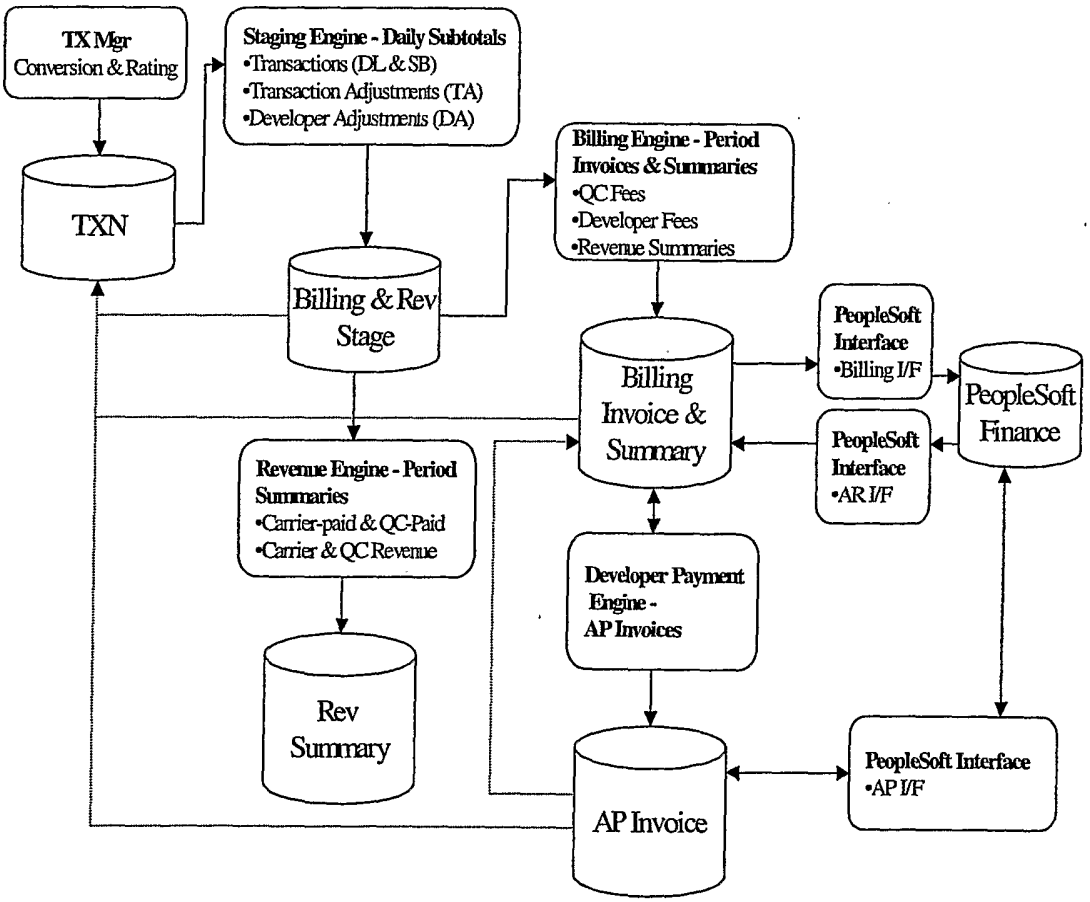
The BREW billing services interfaces to the TX manager, which is the source of the Billing Events (Conversion and Rating) and the MINMAP repository. This is the primary source which drives the billing processing. BREW billing consists of four engines: a staging engine, a carrier invoice engine, and a developer payment engine.

- Staging Engine: The staging engine will perform incremental subtotals of the billing events/MINMAP data through a daily scheduled job. Processing will include computing monthly subtotals for developer fees, revenue share, and developer payment. The types of transactions that are processed are Downloads (DL), Subscriptions (SB), and Adjustments (Transaction adjustments (TA) and Developer adjustments (DA).
- Billing Invoice Engine: The billing invoice engine will derive the carrier invoice subtotals by Developer and billing period from the "staged" data based on QC fees, developer fees. Revenue summaries are processed by manufacturer, sub carrier and part number for each invoice (developer fee (DF) and revenue share (RS).
- Revenue Engine: The revenue engine will derive carrier revenue summaries based on carrier-paid or QC-paid transactions and revenue type.
- Developer Payment Engine: The Developer payment engine will derive the developer payment summaries by part number and carrier from the "staged" data. The developer payment is based on the developer payment period and the developer payment plan (i.e., premium or standard).

Both the invoice Developer summary and the part number summary are stored in the billing summary data. There are three interfaces between PeopleSoft and BREW Billing include:

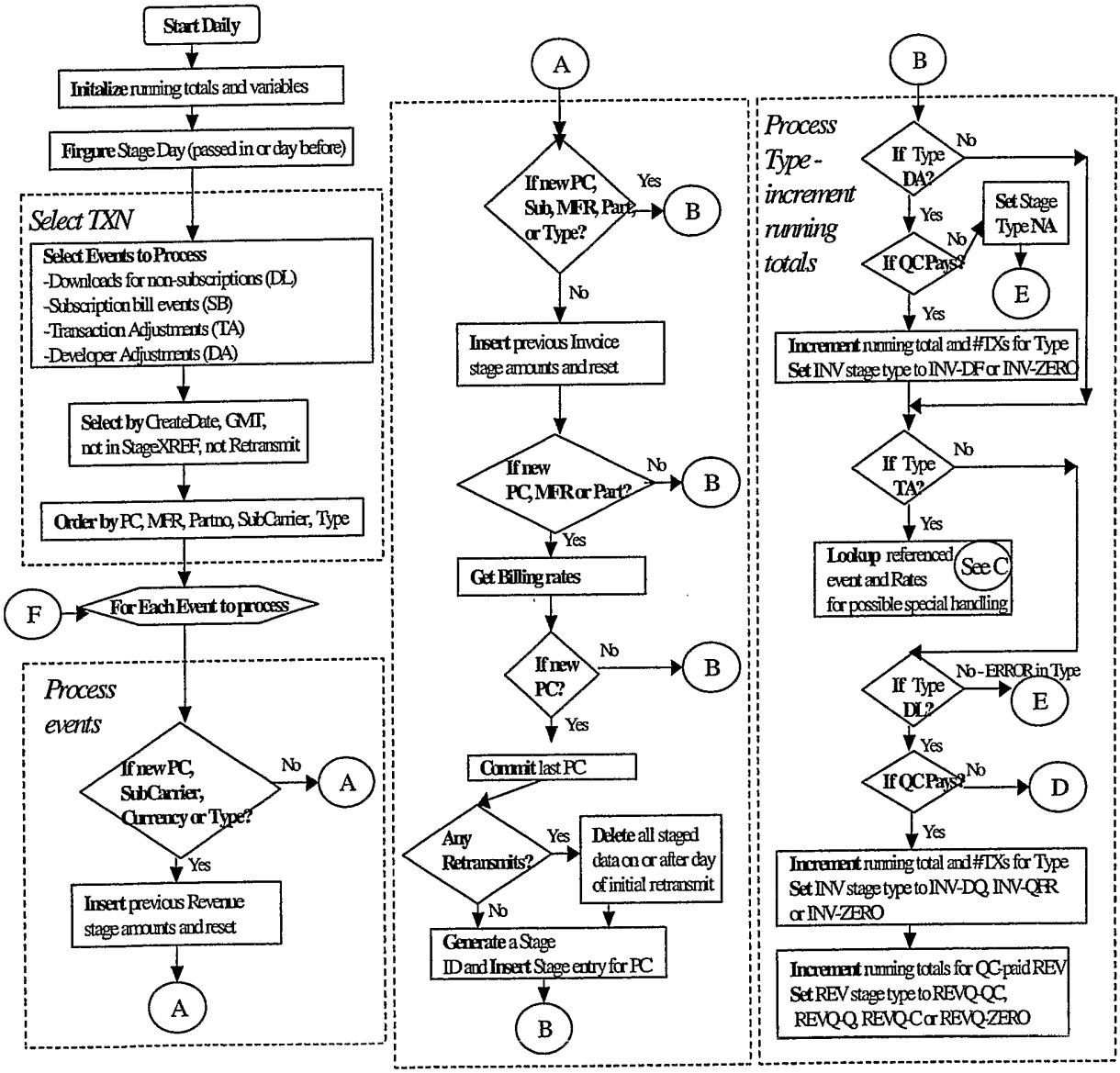
- AR Interface: the AR interface processes payment received from the carriers and maps the payment back to the Developer summary and Billing event data. This is needed in order to drive the developer payment appropriately. For example, in the basic payment plan, a developer would not get paid until QC has received payment from the carrier. As such, only billing events tagged as "paid" would be processed for developer payment.
- Billing Interface: this interface relates to carrier invoicing. BREW billing will generate a billing period invoice total and Developer summaries for each billing period. The Billing interface will query for the invoice total and trigger the generation of the invoicing process through PeopleSoft billing. PeopleSoft will generate the carrier invoice containing the invoice total. If the carrier requires the Developer summary information, this will be provided to the carrier from data in BREW billing. PeopleSoft will not store the Developer summary information.

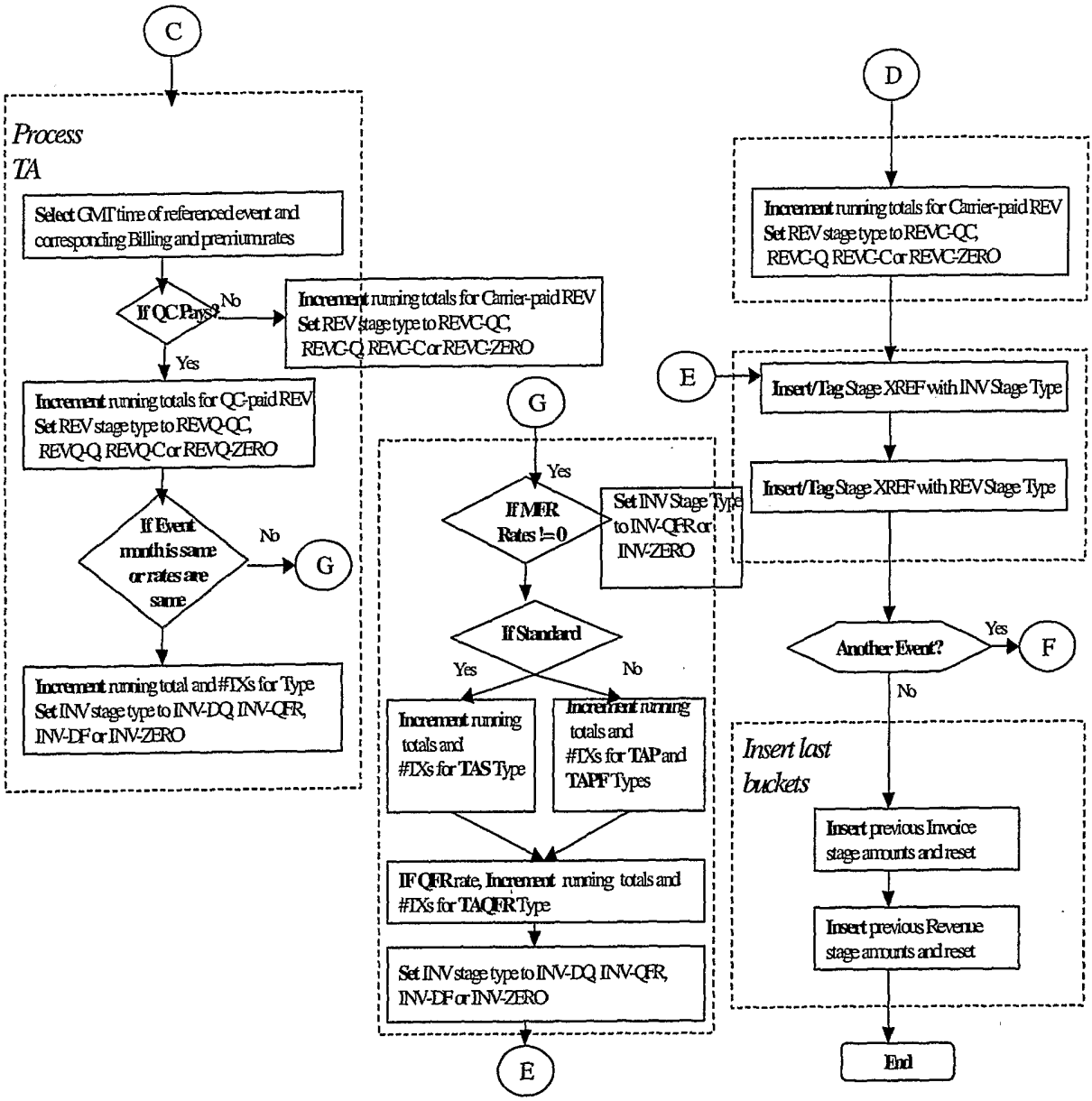
- AP Voucher Interface: this interface processes monthly payment to developers. BREW billing generates the developer payment totals and part number summaries by carrier for each payment period. The AP voucher interface will query for the payment total and trigger the generation of an AP voucher in PeopleSoft. If the developer requires the part number summary information by carrier, this will be provided to the developer from data in BREW billing. PeopleSoft will not store the part number summary information by carrier.



2.1 Staging Engine

The following diagram depicts a high-level logic flow for the daily staging of TXN billing events into daily staging tables at the lowest level (sub-carrier, manufacturer, part number, type, bucket level) per carrier. Events are tagged with stage Ids and stage types.




C
Process TA
Select GMT time of referenced event and corresponding Billing and premium rates
If QC Pays?
No
Yes
Increment running totals for Carrier-paid REV Set REV stage type to REVC-QC, REVC-Q, REVC-C or REVC-ZERO
Increment running totals for QC-paid REV Set REV stage type to REVQ-QC, REVQ-Q, REVQ-C or REVQ-ZERO
If Event month is same or rates are same
No
G
Increment running total and #IXs for Type Set INV stage type to INV-DQ, INV-QFR, INV-DF or INV-ZERO
G
Yes
If MFR Rates != 0
Set INV Stage Type to INV-QFR or INV-ZERO
If Standard
Yes
No
Increment running totals and #IXs for TAS Type
Increment running totals and #IXs for TAP and TAPF Types
IF QFR rate, Increment running totals and #IXs for TAQFR Type
Set INV stage type to INV-DQ, INV-QFR, INV-DF or INV-ZERO
E
D
Increment running totals for Carrier-paid REV Set REV stage type to REVC-QC, REVC-Q, REVC-C or REVC-ZERO
E
Insert/Tag Stage XREF with INV Stage Type
Insert/Tag Stage XREF with REV Stage Type
Another Event?
Yes
F
No
Insert last buckets
Insert previous Invoice stage amounts and reset
Insert previous Revenue stage amounts and reset
End

179
180
181

181

Stage Types

NA	Stage type used to tag events which are skipped because they do not apply to invoices or revenue - example: Carrier-paid DA type
INV-DQ	Invoice stage type used to tag events which apply to both Developer Fee (DF) and QUALCOMM Fee Revenue Share (QFR) invoices
INV-DF	Invoice stage type used to tag events which apply to a Developer Fee (DF) invoice only
INV-QFR	Invoice stage type used to tag events which apply to a QUALCOMM Fee Revenue Share (QFR) invoice only
INV-ZERO	Invoice stage type used to tag events which do not apply to either invoices because the amount is zero or the MFR and QC revenue rates are zero
REVC-QC	Revenue stage type used to tag Carrier-paid events which apply to both QUALCOMM revenue and Carrier Revenue
REVC-Q	Revenue stage type used to tag Carrier-paid events which apply to QUALCOMM revenue only (Carrier rate must be zero)
REVC-C	Revenue stage type used to tag Carrier-paid events which apply to Carrier revenue only (QC rate must be zero)
REVC-ZERO	Revenue stage type used to tag Carrier-paid events which no not apply to Carrier revenue or QUALCOMM revenue because the amount is zero or the Carrier and QC revenue rates are zero
REVQ-QC	Revenue stage type used to tag QC-paid events which apply to both QUALCOMM revenue and Carrier Revenue
REVQ-Q	Revenue stage type used to tag QC-paid events which apply to QUALCOMM revenue only (Carrier rate must be zero)
REVQ-C	Revenue stage type used to tag QC-paid events which apply to Carrier revenue only (QC rate must be zero)
REVQ-ZERO	Revenue stage type used to tag QC-paid events which no not apply to Carrier revenue orQUALCOMM revenue because the amount is zero or the Carrier and QC revenue rates are zero

182

183

Bucket Types

DL	Consolidated non-subscription Download (TXN.Type=DL Method!=3) and Subscription Bill (TXN.Type=SB) events - AMT = TXN.DAP if QC-paid, TXN.PRICE if Carrier-paid
DA	Consolidated Developer Adjustment (TXN.Type=DA) events - AMT = TXN.ADJUSTMENT_PRICE (affects Developer Fee (DF) Invoice only
TA	Consolidated Transaction Adjustment (TXN.Type=TA) events - AMT = TXN.ADJUSTMENT_DAP if QC-paid, TXN.ADJUSTMENT_PRICE if Carrier-paid (applies to ALL TA events for Carrier-paid, applies to all TA events with the same rates as current month for QC-paid
TAS	Consolidated Transaction Adjustment (TXN.Type=TA) events where referenced eventid (TXN.ADJUSTMENT_EVENTID) has different billing and/or premium rate(s) and Manufacturer is Standard - AMT = TXN.ADJUSTMENT_DAP (applies to Developer Fee (DF) Invoice only
TAP	Consolidated Transaction Adjustment (TXN.Type=TA) events where referenced eventid (TXN.ADJUSTMENT_EVENTID) has different billing and/or premium rate(s) and Manufacturer is Premium - AMT = TXN.ADJUSTMENT_DAP (applies to Developer Fee (DF) Invoice only
TAPF	Consolidated Premium Fee for Transaction Adjustment (TXN.Type=TA) events where referenced eventid (TXN.ADJUSTMENT_EVENTID) has different billing and/or premium rate(s) and Manufacturer is Premium(applies to Developer Fee (DF) Invoice only

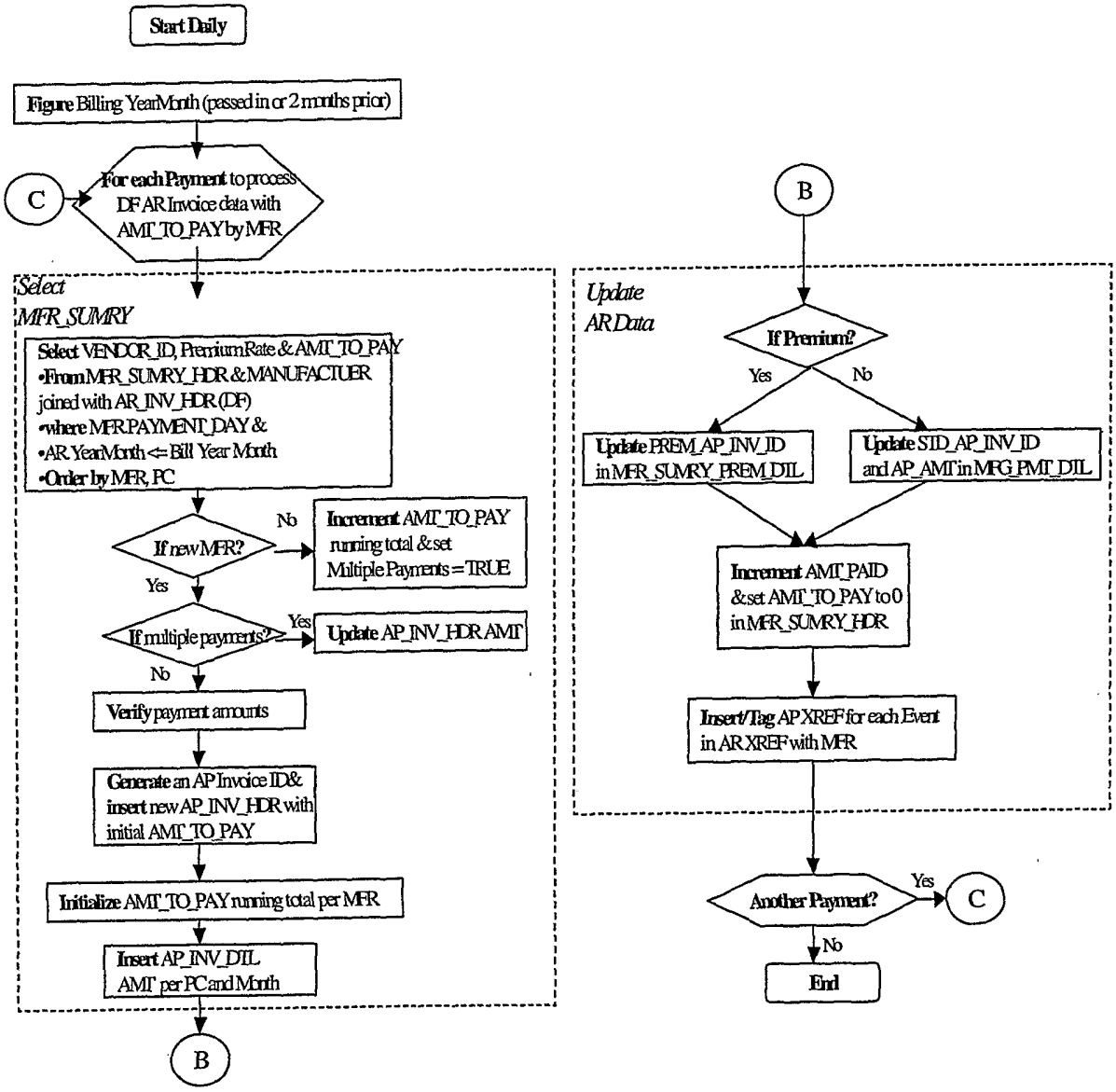
TAQFR	Consolidated Transaction Adjustment (TXN.Type=TA) events where referenced eventid (TXN.ADJUSTMENT_EVENTID) has different billing and/or premium rate(s) - AMT = TXN.ADJUSTMENT_DAP (applies to QUALCOMM Fee Revenue Share (QFR) Invoice only

184

185

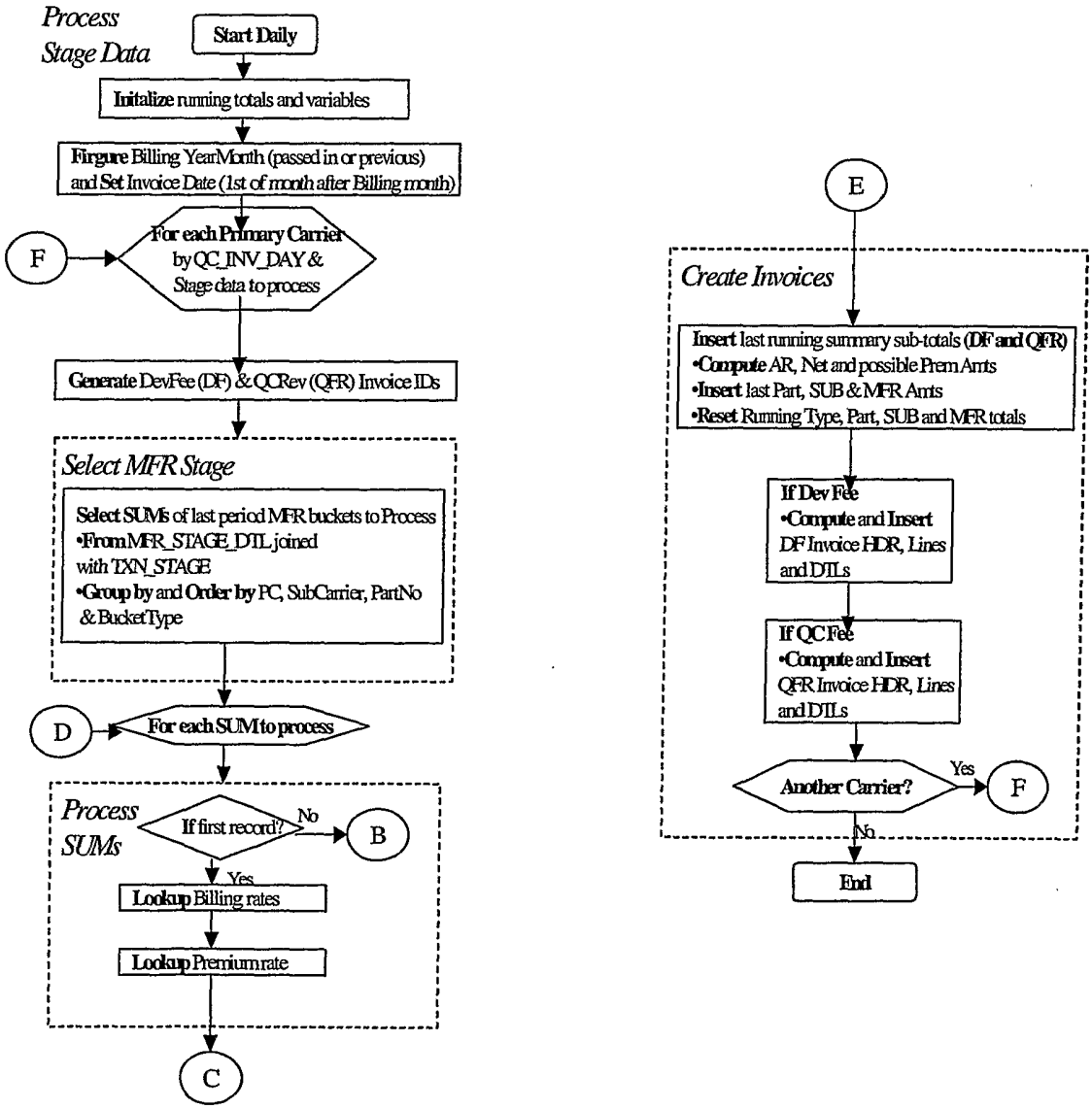
2.2 Payment Engine

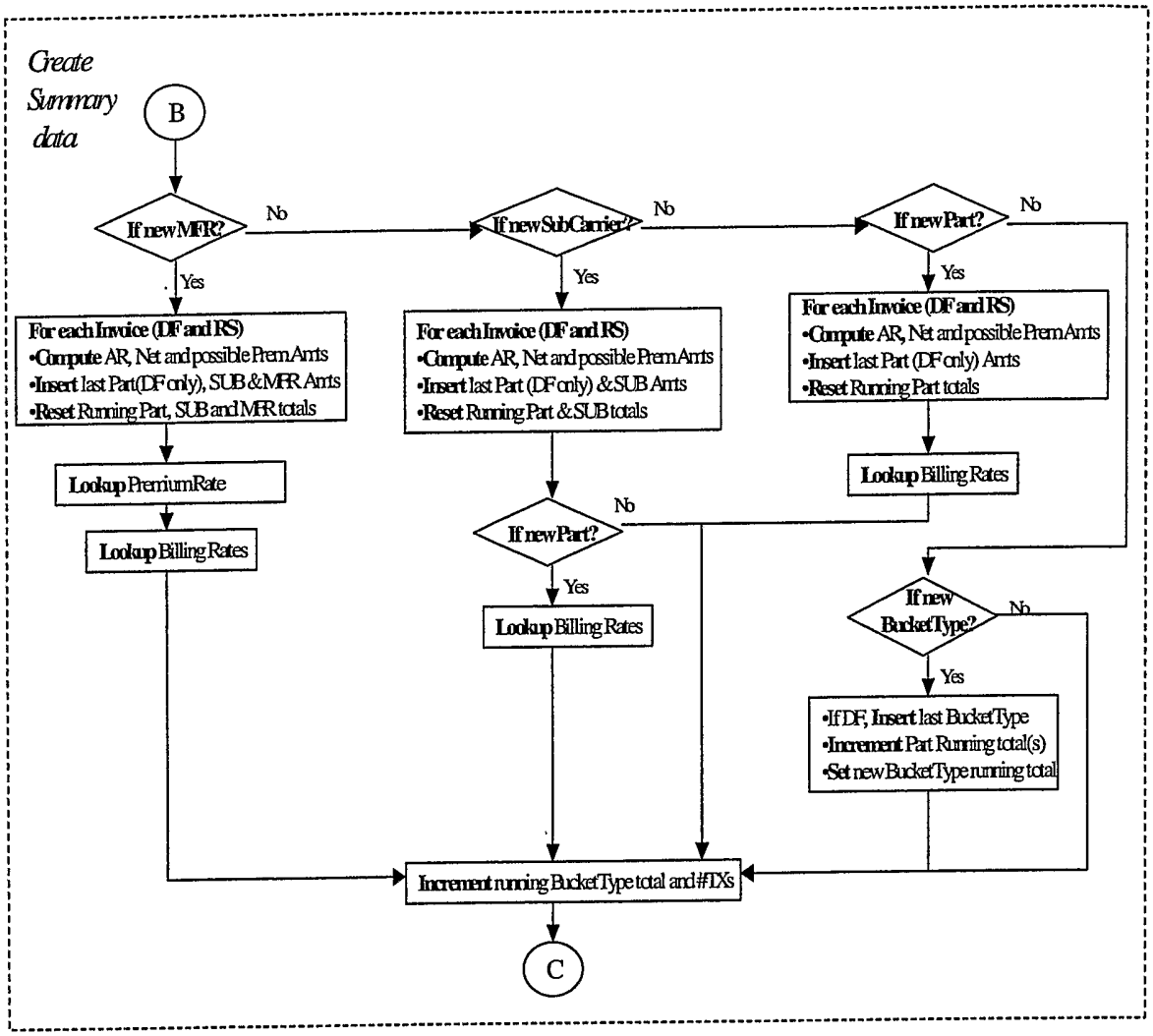
The payment engine will process the monthly creation of AP Invoices per Manufacturer from AR Invoice and payment data. The developer payment is based on the payment period for premium or standard developers.



2.3 Billing Engine

The billing engine will be computing monthly subtotals for each primary carrier and will generate developer fee and revenue share invoice totals tags each event with invoice IDs. Processing also includes consolidation for invoice lines and AR buckets; and manufacturer/subcarrier (invoices) and part summaries (developer fees only).





199
200

201 **AR Invoice Line Types**

DF	Developer Fee
DFA	Developer Fee Adjustment
QFR	Qualcomm Fee Revenue Share
QFRA	Qualcomm Fee Revenue Share Adjustment
QFE	Qualcomm Fee Enablement Fee
QFEA	Qualcomm Fee Enablement Adjustment

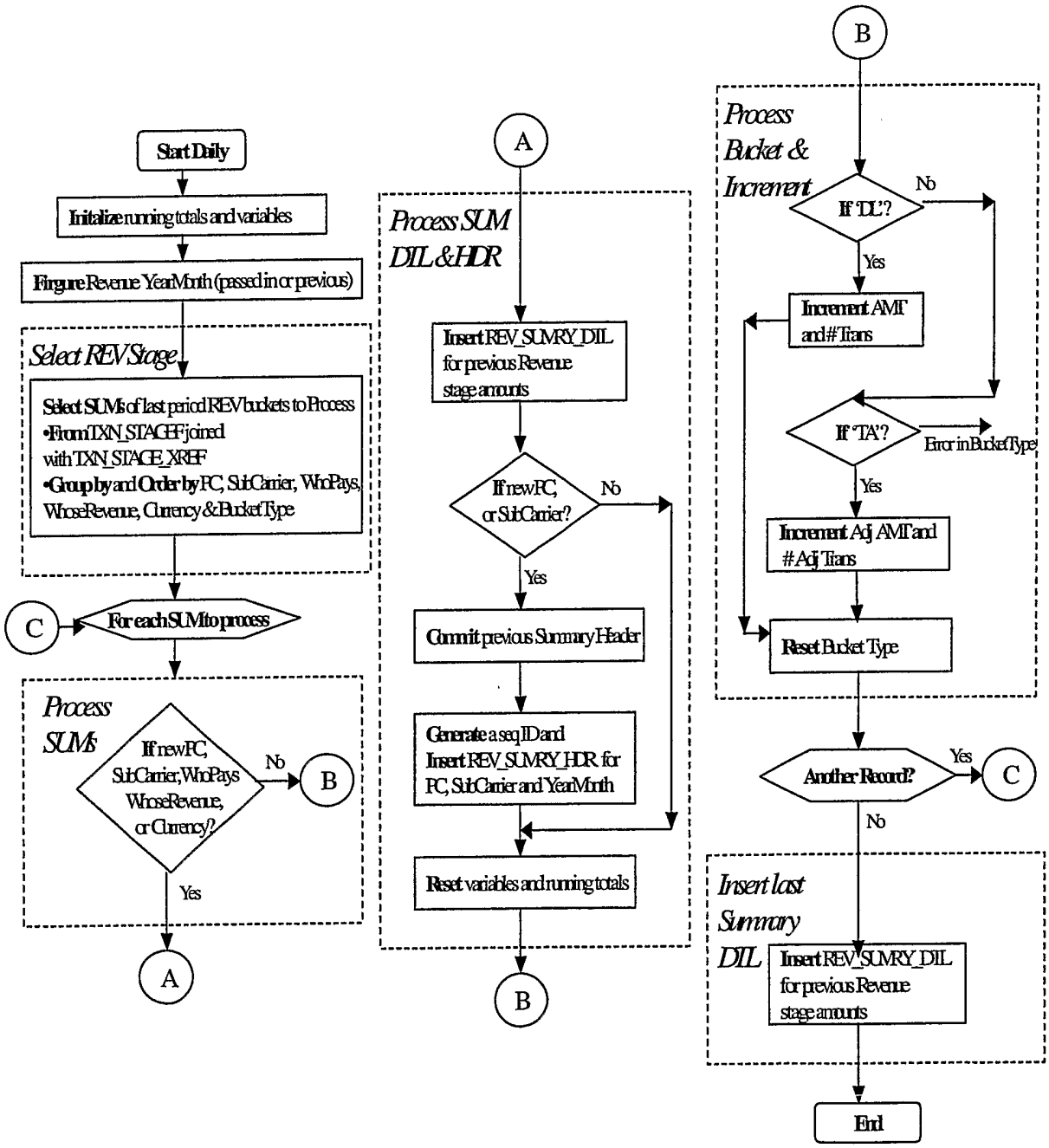
202

203

204

2.4 Revenue Engine

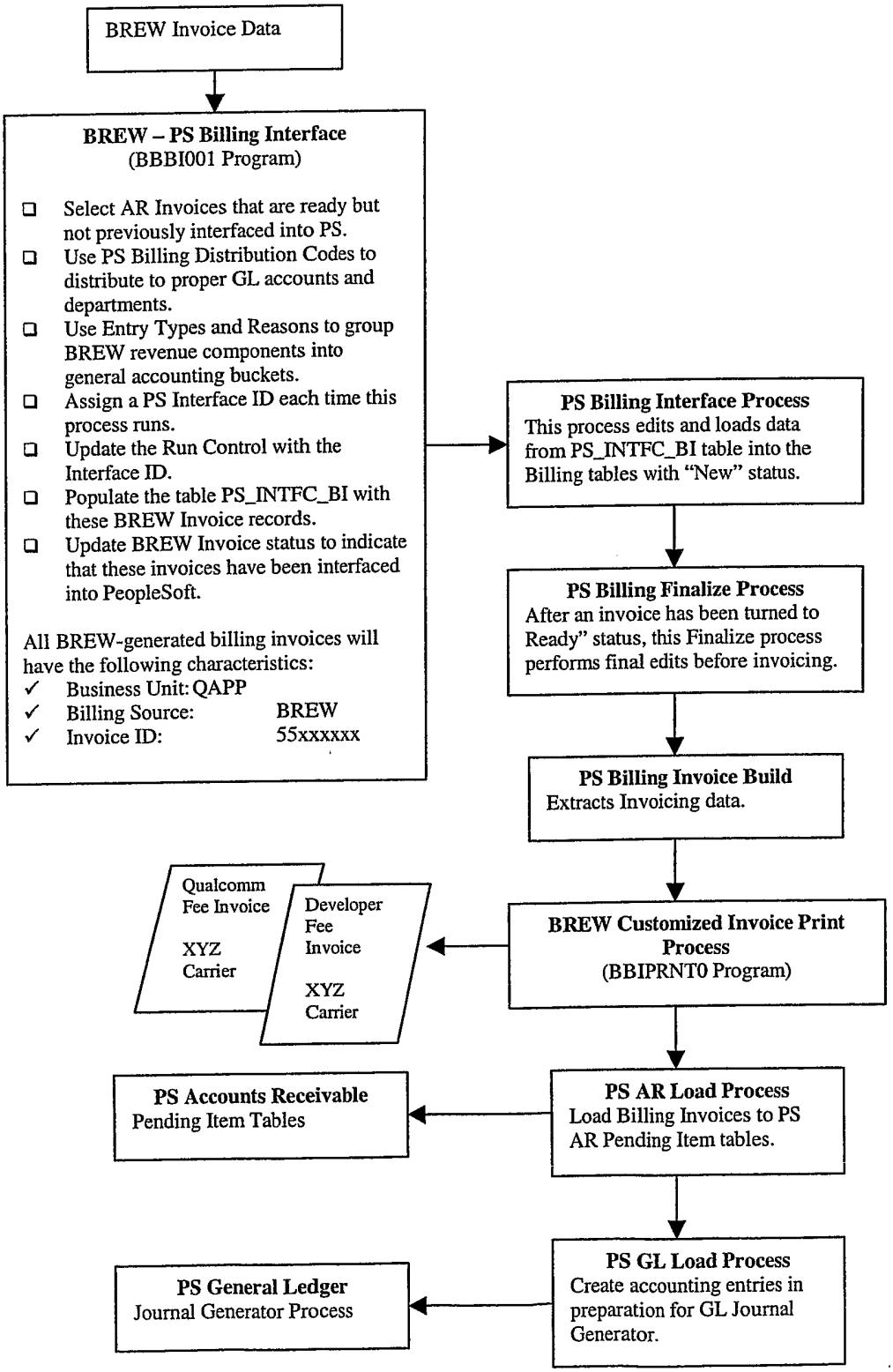
The revenue engine will derive the carrier revenue subtotals by Developer and billing period from the "staged" data for carrier reporting.



209

210 ## *2.5 PeopleSoft Interfaces*

211 ### 2.5.1 Billing



212

213

214 The new BREW program BBBI001 will read data from BREW Billing AR Invoice tables and
215 populate the PeopleSoft Billing Interface tables based on the following mapping. Besides these
216 fields coming from BREW Billing, there are numerous others that require default values for
217 PeopleSoft which this interface program will populate appropriately.

218

219

BREW Billing Source Data	PeopleSoft Target Data PS_INTFC_BI Table
AR_INV_HDR. STATUS = 'R' (Ready Status)	BUSINESS_UNIT = 'QAPP'
AR_INV_HDR. CUST_ID	BILL_TO_CUST_ID
AR_INV_HDR. INV_TYPE	BILL_TYPE_ID
AR_INV_HDR. CREATED_DATE	INVOICE_DT
AR_INV_HDR. AR_INV_ID	TARGET_INVOICE
LINE_TYPES. DSC	DESCR
AR_INV_HDR. AMT	GROSS_EXTENDED_AMT
AR_INV_LINE_DTL. (ENTRY_TYPE + '-' + ENTRY_REASON)	DST_ID
AR_INV_HDR. YEAR_MONTH	FROM_DT, TO_DT
AR_INV_LINE_DTL. ENTRY_TYPE	ENTRY_TYPE
AR_INV_LINE_DTL. ENTRY_REASON	ENTRY_REASON

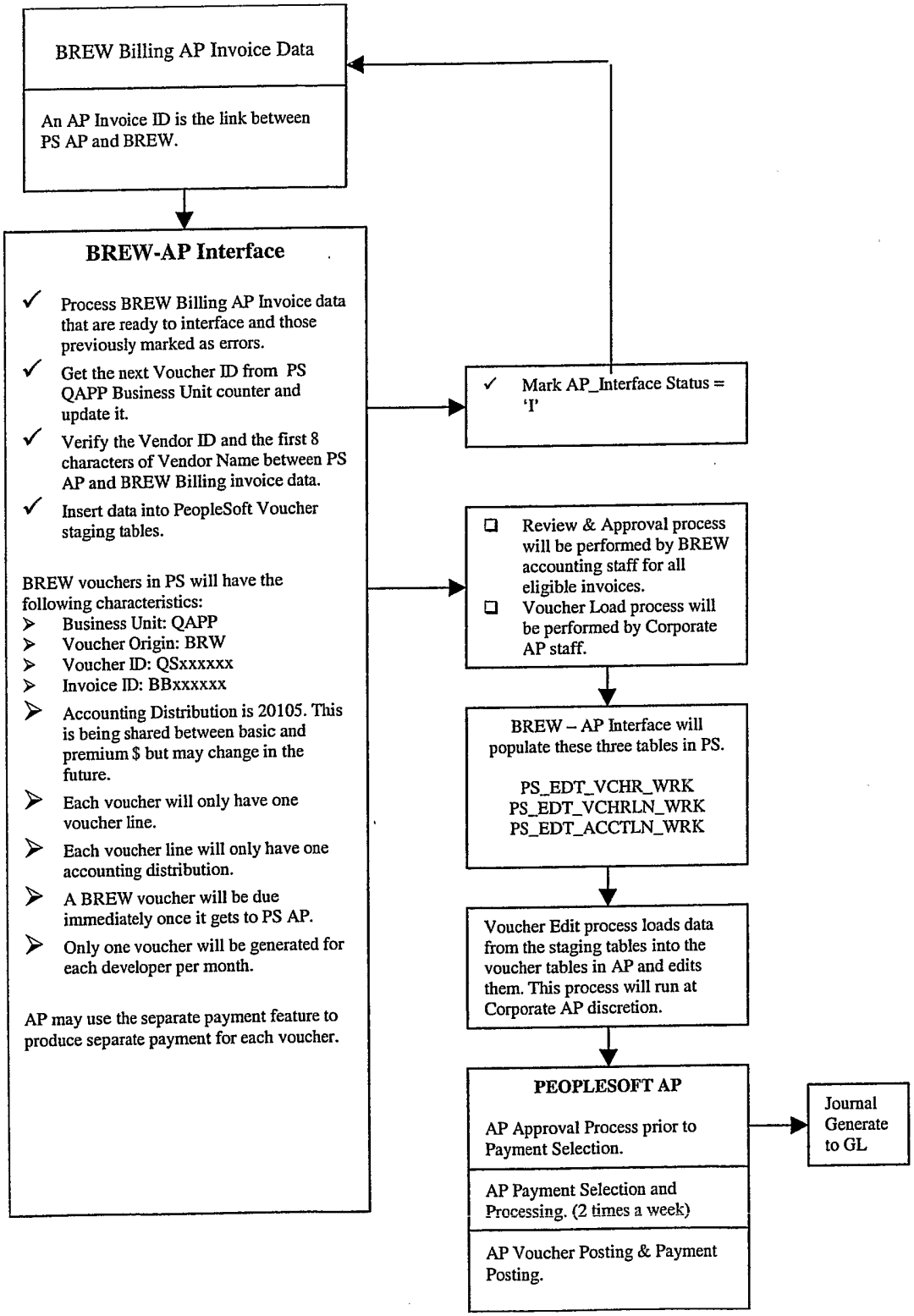
220

221

222

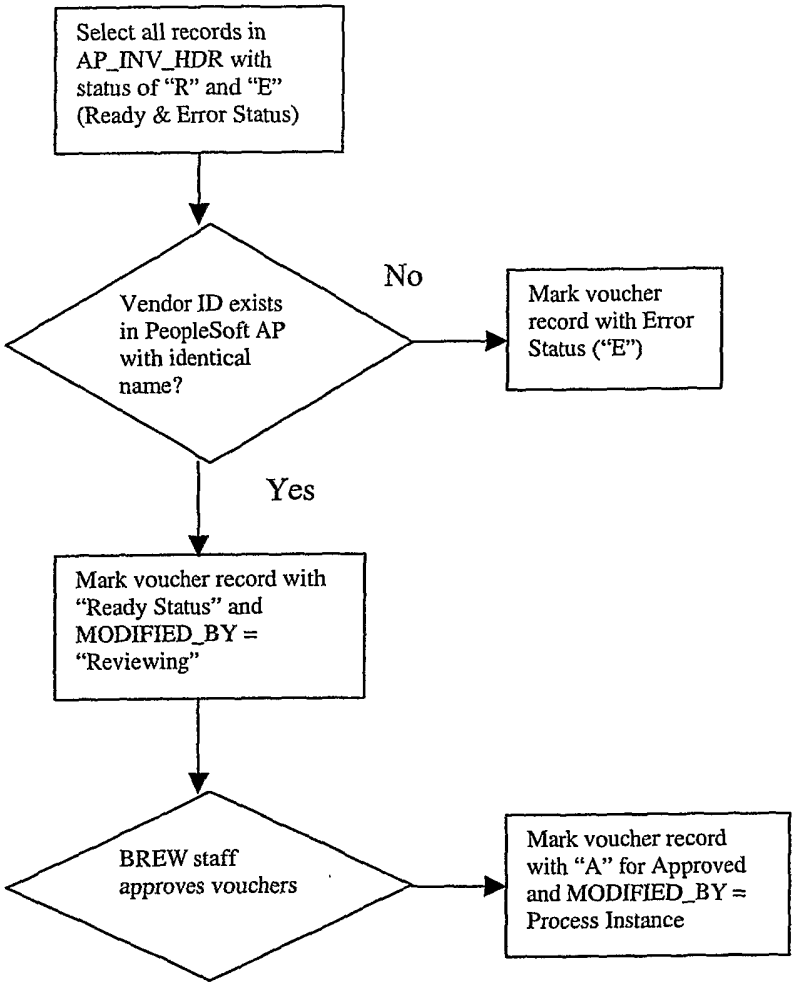
BREW Billing Source Data	PeopleSoft Target Data PS_INTFC_BI_NOTE Table
AR_INV_HDR. STATUS = 'R' (Ready Status)	BUSINESS_UNIT = 'QAPP'
AR_INV_HDR. AR_INV_ID	TARGET_INVOICE
AR_INV_HDR. DSC	TEXT254 (Example: Total Downloads = 14)

223 ***2.5.2 Accounts Payable***



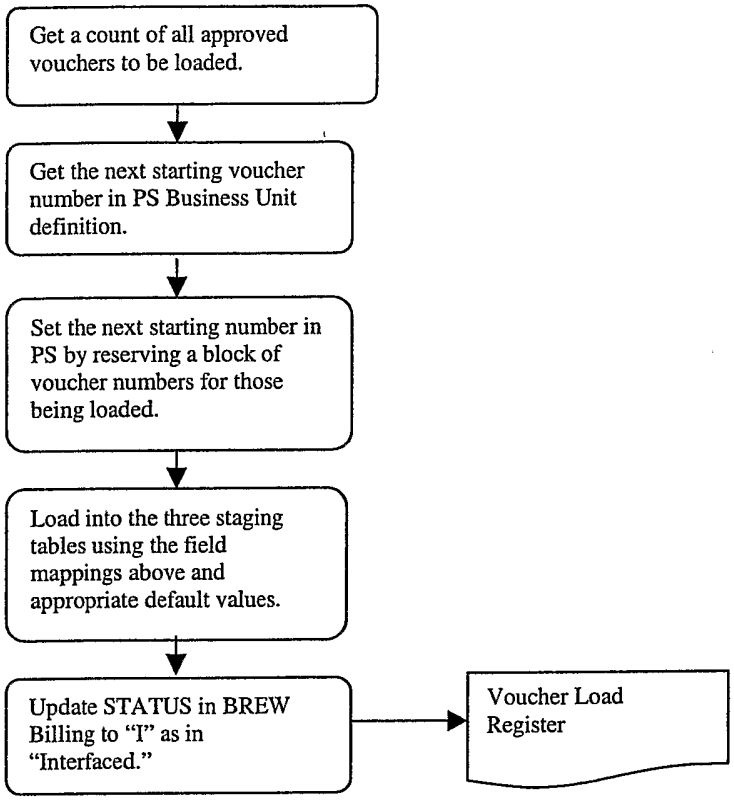
224
225
226

BREW Billing Review and Approval Process: This is a procedure that BREW accounting staff need to approve incoming vouchers and authorize Corporate AP to load, edit then pay them. Once approved, the report created by this process serves as a hard copy of the authorization that will be sent to Corp AP.



226
227
228

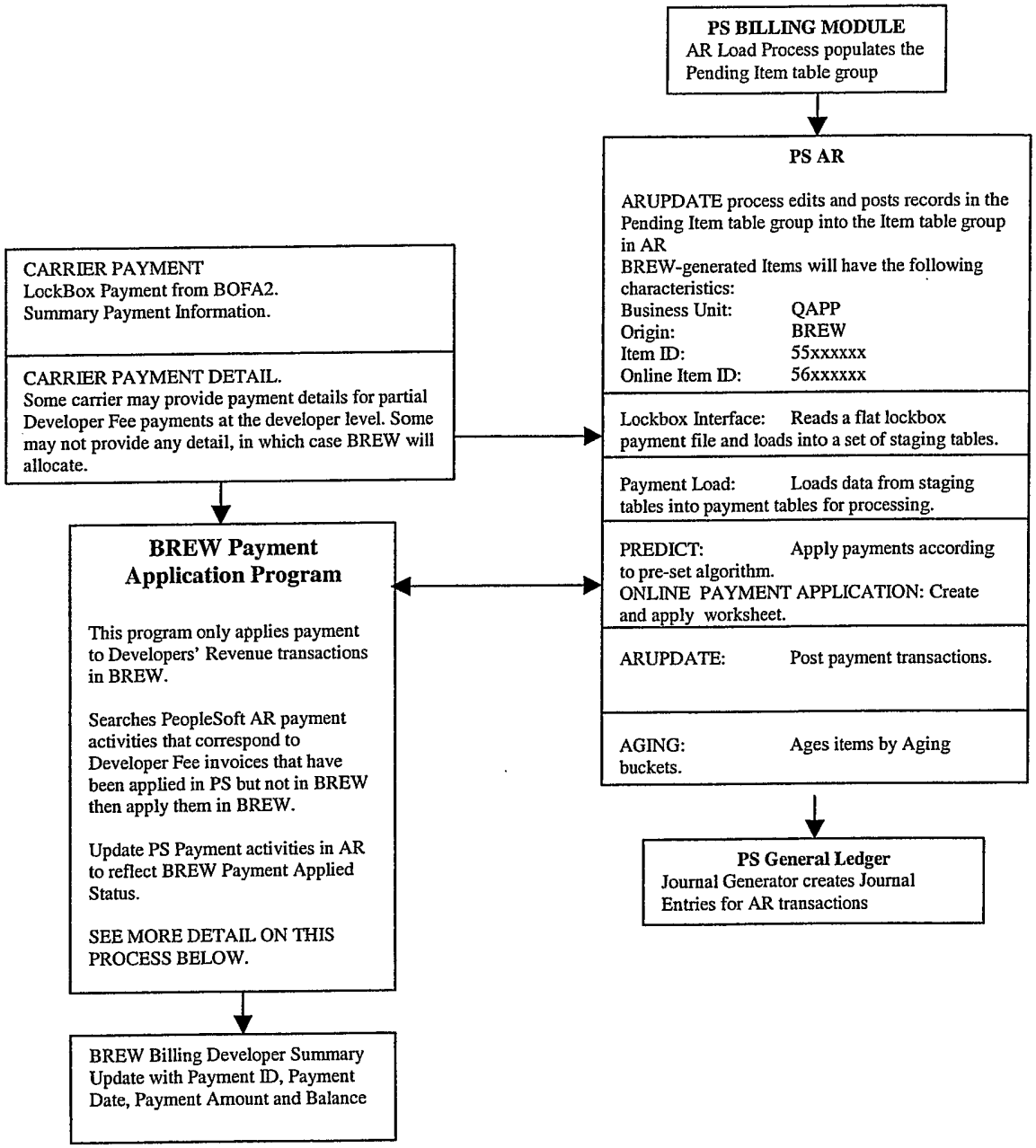
Voucher Load Process: Once a voucher has been approved by BREW, it is eligible for the next process to load into PeopleSoft voucher staging tables.



228
229

230 ### 2.5.3 Accounts Receivable

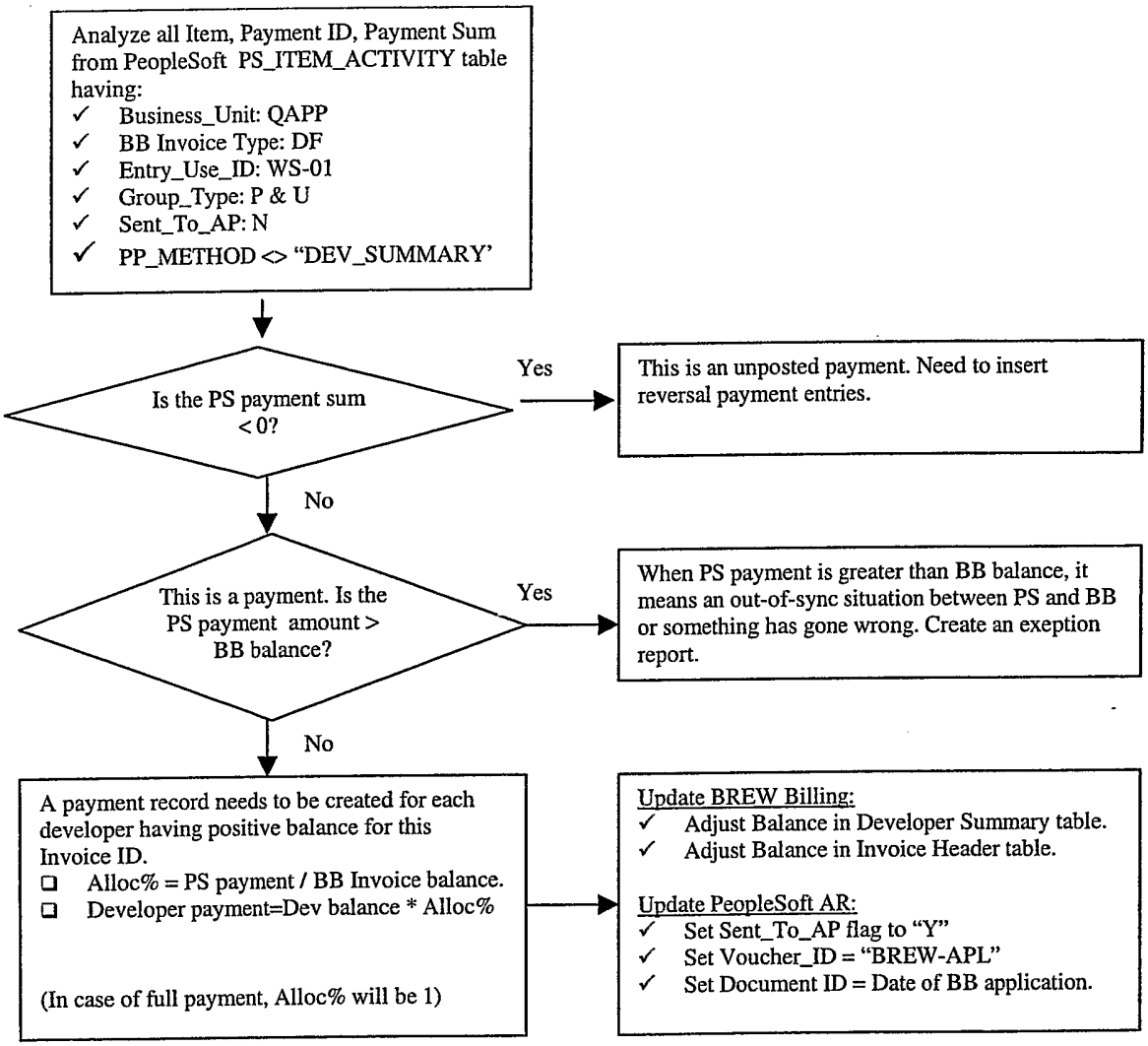
231 ### AR Overview



232
233
234

234 ***Payment Allocation***

235



236
237
238

238

Payment by Carrier Detailed File

Read input file(s) from a carrier or multiple carriers.

- A file can contain multiple payments, each of which can pay for more than one invoice. At the carrier, payment id and Invoice id combination level, BREW requires one header and at least one detail record.
- The header record contains Customer Id, Payment ID and Payment amount.
- The detail record contains Manufacturer (developer) ID and payment amount for that developer.

	Position	Length
Record Type (1 for header)	1	1
Customer ID (CUST_ID)	5	15
Invoice ID (INVOICE)	20	20
Payment ID (PAYMENT_ID)	50	20
Payment Amount (No significant)	80	20

	Position	Length
Record Type (2 for detail)	1	1
Manufacturer/Developer Name	5	35
Payment Amount for Developer	50	20

Header Record Edits:

- ✓ Validate Customer as one set up for Developer Summary input file.
- ✓ Validate Payment ID in the input file as one posted and not BREW-applied before.
- ✓ Validate Invoice ID as an Item ID in PS AR.
- ✓ Validate payment amount to be numeric and equal to the payment amount posted in PS for that Item and Payment ID.
- ✓ Check for negative payment amount.
- ✓ Check for non-DF data.

Detail Record Edits:

- ✓ Double check the total of amounts for each Invoice ID and Payment ID combination to match that in PS. This also indirectly check detail records against header record.
- ✓ Check for negative amount.
- ✓ Check for overpayment amount.
- ✓ Check against the Invoice Developer Summary for Developer validity on that Invoice ID.

Any Edit failure?

Yes → Delete temporary data from Payment Detail table. Reject the WHOLE file. Goes on to the next input file.

No

- ✓ Update Developer Summary and Invoice Header tables.
- ✓ Update PS_ITEM_ACTIVITY to set Sent_To_AP and Voucher ID accordingly.

239
240

2.5.4 Invoice Print Program

Due to unique business requirements for BREW, we have had to create a new Invoice Print program. Besides the general formatting requirements such as Billing Period, Wire Instruction, and BREW Invoice Terms which are very specific to BREW, this program will also satisfy the following requirements:

A description for each Invoice ID.

An Invoice in PS Billing does not have a description. It simply is a 22-character field. PeopleSoft only provides for description at the Invoice line level.
For BREW, we will use the Bill Type description which is associated with each Invoice to display on the Invoice itself. For example, a BREW Invoice can be created for Qualcomm Enablement Fee. Therefore, it should be associated with Bill Type "QF1". The description "Qualcomm Enablement Fee" in this list box here would then be what's displayed on the hard copy invoice to the carrier.

All Bill Type Identifiers for BREW are set up under QAPP SetID:

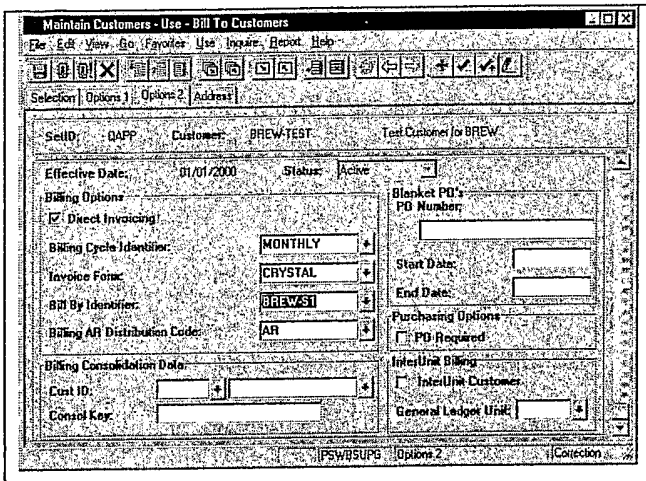
- DF Developer Fees
- QFE Qualcomm Enablement Fee
- QFA BREW Annual Maintenance Fee
- QFL BREW Licensing Fee
- QFR Qualcomm Revenue Share

Particular Invoice Grouping Requirement At The Carrier Level

BREW has a distinctive requirement that for a particular carrier, they may be sent invoice(s) in two ways:

1. One hard copy invoice for each Invoice ID.
2. One hard copy for Qualcomm Enablement Fee and Qualcomm Revenue Share. Another hard copy for Developer Fee.

To achive this objective, BREW will make use of the Bill-By Identifier to process invoices accordingly.
This designation is set at the Customer level in Maintain Customer | Use | Bill To Customer



The Bill-By Identifier is used in PeopleSoft to denote how a billing interface record should be created in PS. Depending on how this field is defined, a new record may be added to existing bills or as new ones.
BREW uses this field slightly different.

- We have set up BREW-S1 which will invoke the BREW Invoice Print program to group Invoice Ids that share the first two characters of the Bill Type. For example, all QF1 and QF4 will be displayed on one single hard copy which will be the Qualcomm Fees Invoice. On the other hand, Invoice with DF Bill Type will be on another hard copy which is the Developer Fee Invoice.
- BREW-ALL will display all Invoices for that carrier to be included on one hard copy.

Anything else will cause the print program to create one invoice hard copy per Invoice ID.

277
278
279

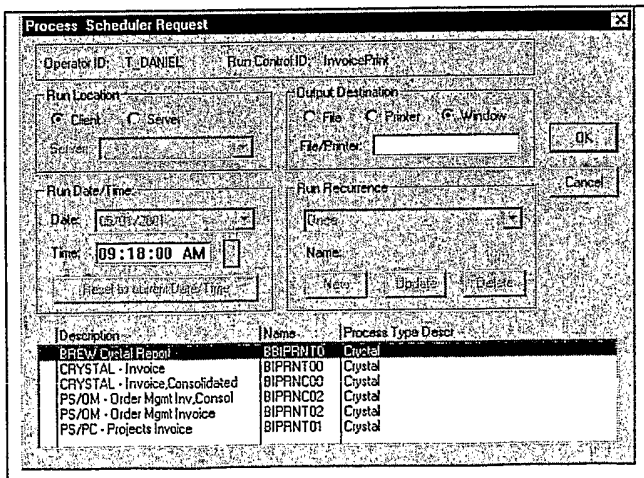
BREW Invoice Processing Procedure

Go to:
Manage Sales Activities | Generate Invoices | Process | Finalize & Print

After an Invoice has been finalized, we can start the process to create BREW invoice for it.

Select the job BIIVCEN (Process Extract Table Invoice.)
Once completed, it would have updated the same Run Control ID with a Process Instance from which the appropriate Invoices would be created. For the next step, use the same Run Control ID and go to …

280



Manage Sales Activities | Generate Invoices | Process | Print Crystal Invoices.

The "From Process Instance" and "To Process Instance" values populated on this panel are the result of the previous job (BIIVCEN). Leave them as they are.

Run Location: Client.
Output Destination: Window
Name: BBIPRNT0 (BREW Crystal Report)

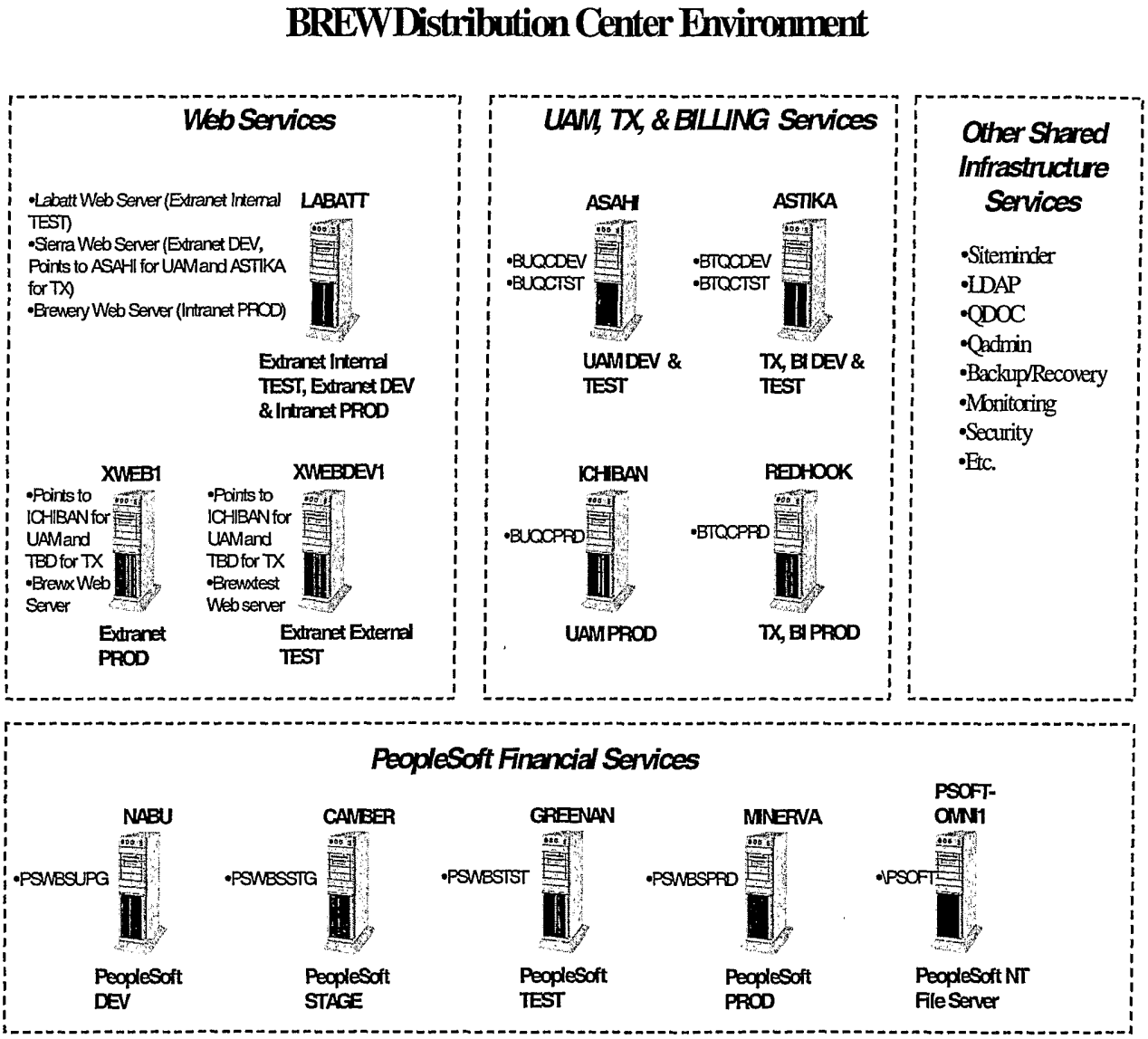
When output is ready to be printed, make sure the printer is loaded with the appropriate Qualcomm paper stock.

281

282

283

283 # 3 System Characteristics



284
285

286

4 Data Model

The Brew Billing Data model can be found in Livelink at:
http://lvcrpprd.qualcomm.com/livelink/livelink/22347763/Brew_Billing_Schema.doc?func=doc.Fetch&nodeid=22347763

Staging Engine:

TXN (TXNMGR)	PRIMARY_CARRIER (TXNMGR)
TXN_STAGE_XREF	MANUFACTURER (TXNMGR)
TXN_STAGE	MFG_PREMIUM (TXNMGR)
MFR_STAGE_DTL	REV_STAGE_DTL

Revenue Engine:

REV_SUMRY_HDR
REV_SUMRY_DTL

Billing Engine:

AR_INV_HDR	PRIMARY_CARRIER (TXNMGR)
AR_INV_LINES	MFR_STAGE_DTL
AR_INV_LINE_DTL	MFG_PREMIUM (TXNMGR)

MFR_SUMRY_HDR
MFR_PART_SUMRY_HDR
MFR_PART_SUMRY_LINES
MFR_PART_SUMRY_DTL
MFR_SUB_SUMRY_PREM_DTL
TXN_AR_XREF

Payment Engine:

MFR_PMT_DTL	TXN_AP_XREF
AP_INV_HDR	AP_INV_DTL

316 # Appendix A - Glossary

317 This glossary defines terms, acronyms, and abbreviations used in the document.

ACC	**Application Certification Center**
ADS	**Application Download Server**
BREW	**Binary Runtime Environment for Wireless**
DSRP	**Developer's Suggested Retail Price**
EFT/ACH	**Electronic File Transfer/ Automated Clearing House**
ESN	**Electronic Serial Number**
GMT	**Greenwich Mean Time**
MIN	**Mobile ID Number**
OEM	**Original Equipment Manufacturer**
QC	**QUALCOMM**
QDC	**QIS Distribution Center**
QIS	**QUALCOMM Internet Services**
SID	**Subscriber Identification Number**
TX	**Transaction**
UAM	**Unified Application Management**
XML	**Extensible Markup Language**

318

Appendix B Invoices/Summaries/Vouchers

Qualcomm Fees Invoice Data

Customer: [redacted] XXX
Invoice Date: February 1, 2001
Payment Terms: Net 30
Due Date: 3/3/01
Period: January 2001

Invoice ID 55000002 **Revenue Share** **$9**

Invoice ID 55000003 **Enablement Fee** **$14**

Total downloads: 7

Revenue Share Summary – Carrier Invoice Report

Sub-Carrier	Manufacturer	Revenue Share Subtotal	Adjustment Subtotal	Net Revenue Share
[redacted]1-XXX	DEV-1	$3	<$1>	$2
[redacted]1-XXX	DEV-3	$4	0	$4
[redacted]2-XXX	DEV-3	$3	0	$3
		$10	**<$1>**	**$9**

First time Download/Pre-install Events

Sub-Carrier ID	Event ID	SID	Event Date	Create Date	Part #	Part Name	Enablement Fee
[redacted]-1-XXX	E1	5551111	1/01/01	/1/01/01	P1	Pokemon	$2
[redacted]1-XXX	E8	5058888	1/03/01	1/03/01	P4	Mario Bros	$2
[redacted]1-XXX	E9	505888	1/03/01	1/03/01	P5	Centipede	$2

		8					
[redacted]1-XXX	E10	5058888	1/03/01	1/03/01	P1	Pokemon	$2
[redacted]2-XXY	E2	5652222	1/01/01	1/01/01	P2	[redacted] Calendar	$2
[redacted]-2-XXX	E5	5355555	1/02/01	1/02/01	P3	NASCAR	$2
[redacted]2-XXX	E12	6066666	1/03/01	1/03/01	P4	Mario Bros	$2

340

Developer Fee Invoice Data

Customer: [redacted] XXX
Invoice ID: 55000001
Invoice Date: February 1, 2001
Payment Terms: Net 30
Due Date: 3/3/01
Period: January 2001

Developer Fee Total: **$84**

Developer Fee Summary – Carrier Invoice Report

Sub-Carrier	Manufacturer	Manufacturer Fee Subtotal	Adjustment Subtotal	Net Manufacturer Fee
[redacted]-1-XXX	DEV-1	$24	<$3>	$21
[redacted]-1-XXX	DEV-3	$36	0	$36
[redacted]-2-XXX	DEV-3	$27	0	$27
		$87	**<$3>**	**$84**

Manufacturer Payment Voucher Data

Vendor: DEV-1
Check #: 123
Payment Date: March 18, 2001
Invoice ID: BB000004

Payment Amount: $20.20

Manufacturer Payment Summary – Manufacturer Payment Report

(Premium Developers will get full payment minus premium fee)

For Dev-1

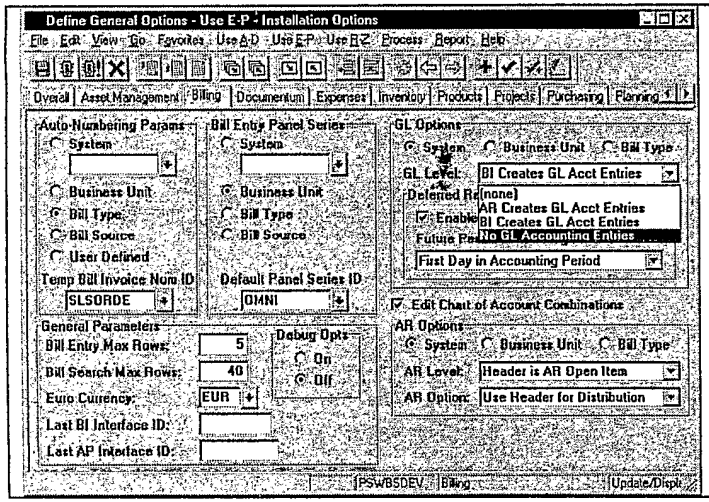
Billing Period	Primary Carrier	Payment Amount
January 2001	[redacted] XXX	$20.20
		$20.20

For Dev-3

Billing Period	Primary Carrier	Payment Amount
January 2001	[redacted] XXX	$52.50
		$52.50

Appendix C PeopleSoft Configuration

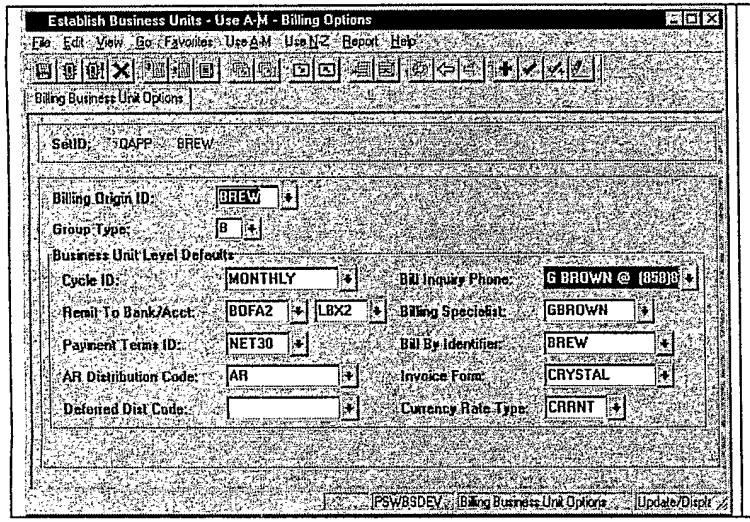
PeopleSoft Billing



BREW will use the existing setup in the current production environment.

- ✓ **Billing Business Unit:** QAPP
- ✓ **Billing Currency:** USD
- ✓ **Journal Template:** BI_BILLING.
- ✓ **Location Code:** Qualcomm

- ☐ **Tax Controls:** None.
- ☐ **InterUnit Voucher Control:** Not Applicable.



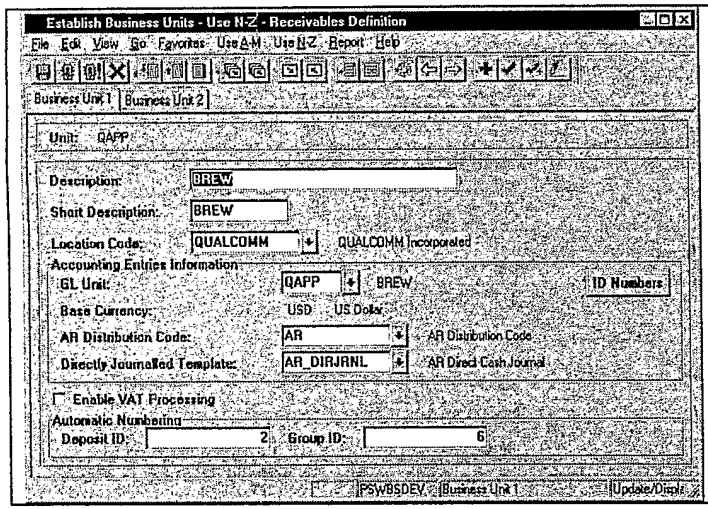
- ✓ **Cycle ID:** MONTHLY
- ✓ **Bank Account:** BOFA2 – LBX2
- ✓ **Invoice Form:** CRYSTAL

More PS Billing Setup Issues

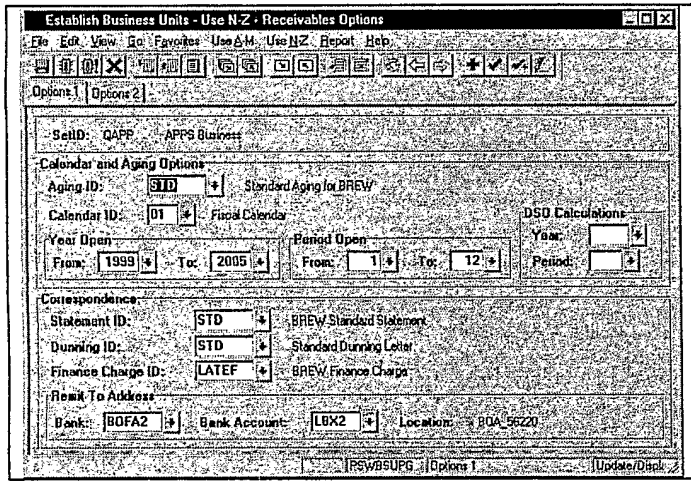
- **Bill-by Identifier:** BREW, BREW-S1
- **Bill Inquiry Phone:** GENE BROWN
- **Bill Source:** BREW
- **Bill Type:** DF, QFR, QFE, QFA, QFL
- **Billing Specialist:** A CHAN, GBROWN

- **Invoice Number ID**: BREW (56xxxxxx)
- **Origin**: BREW
- **Region Types:** BREW. (Set up in Establish Order Processing | Use N-Z | Region Codes.)
- **Region Codes:** SGMT and ACCT
- **Region Category ID:** BREW.

PeopleSoft Accounts Receivable

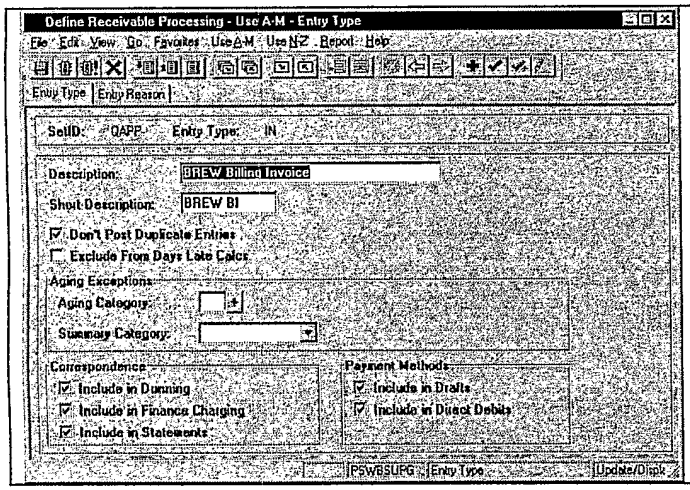


- ✓ **Name of the new AR Business Unit:** QAPP
- ✓ **GL Business Unit:** QAPP
- ✓ **AR Distribution Code:** AR
- ✓ **Directly Journal Template:** AR_DIRJRNL



QAPP will use its own QAPP option and parameters.

- ✓ **Aging ID:** STD
- ✓ **Calendar ID:** 01
- ✓ **Statement ID:** STD
- ✓ **Dunning ID:** STD
- ✓ **Finance Charge:** STD
- ✓ **Remit Bank Information:** BOFA2 – LBX2



See tables below.

User-Defined Entry Types:

SetID	Entry Type	Reason	Acct	DeptID	Prod
QAPP	CR	BADDB	11152	5106	
QAPP	CR	DF	20105	5106	
QAPP	CR	ENFE	11155	5106	ENFE
QAPP	CR	ERROR	11155	5106	
QAPP	CR	MISAP	10122	5106	
QAPP	CR	OVRPY	11155	5106	
QAPP	CR	RECON	11155	5106	
QAPP	CR	REVQ	11155	5106	REVQ
QAPP	CR	REVQP	11155	5106	REVQP
QAPP	DR	DF	20105	5106	
QAPP	DR	ENFE	41100	5106	ENFE
QAPP	DR	FNCHG	48050	5106	
QAPP	DR	REVQ	41100	5106	REVQ
QAPP	DR	REVQP	41100	5106	REVQP
QAPP	IN	AMFEE	41100	5106	AMFEE

QAPP	IN	AUTHE	41100	5106	AUTHE
QAPP	IN	CERT	41100	5106	CERT
QAPP	IN	DF	20105	5106	
QAPP	IN	ENFE	41100	5106	ENFE
QAPP	IN	LICFE	41100	5106	LICFE
QAPP	IN	REVQ	41100	5106	REVQ
QAPP	IN	REVQP	41100	5106	REVQP
QAPP	TAXCR	CA	26511	5106	
QAPP	TAXCR	GA	26515	5106	
QAPP	TAXCR	MO	26514	5106	

Auto Entry Types:

SetID	Function	Entry Type	Reason	Acct	DeptID	Prod
QAPP	FC-01	FC		48050	5106	
QAPP	MT-02	WOC	ADJ	11155	5106	
QAPP	MT-03	WO	ADJ	11155	5106	
QAPP	WS-02	DE		11155	5106	
QAPP	WS-03	DU		11155	5106	
QAPP	WS-09	WO	ADJ	11155	5106	
QAPP	WS-10	WAO	ADJ	11155	5106	
QAPP	WS-11	WAU	ADJ	11155	5106	

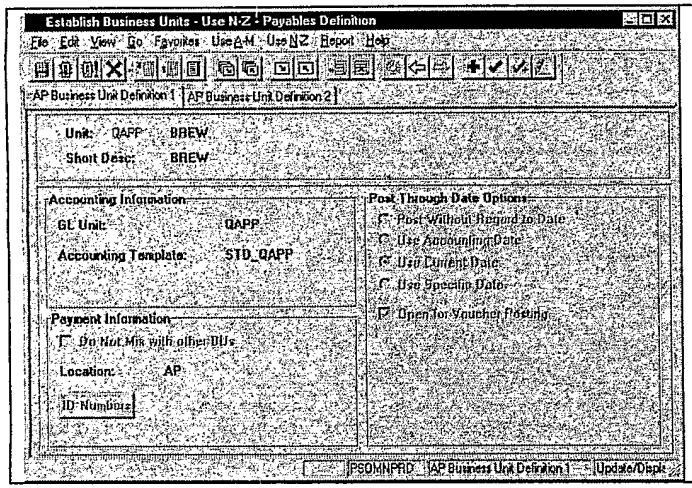
Other AR Setup parameters	
✓ **Aging Definition:** STD ✓ **Collection Status:**CST ✓ **Collector:** ACHAN, GBROWN ✓ **Credit Analyst:** ACHAN, BROWN	✓ **Origin:** BREW. ✓ **Payment Predictor Method:** ALLOCATION & DEVLPR_SUMMARY. ✓ **Sales Person:** ACHAN, GBROWN

✓ **Deposit Type:** Use existing ones under CORP. ✓ **Dispute Status:** BIL (Billing Error) ✓ **Distribution Code:** See table below.	✓ **Statement:** STD ✓ **Dunning:** STD ✓ **Group Type:** B: Billing; C: Credit, PS_AR.

SetID	Code	Descr	Acct	DeptID	Prod
QAPP	AR	AR Distribution Code	11110		
QAPP	CR-DF	Developer Fee Adjustment CR	20105	5106	
QAPP	CR-ENFE	BREW Enablement Fee Adjust CR	11155	5106	ENFE
QAPP	CR-REVQ	Qualcomm Revenue Share Adj CR	11155	5106	REVQ
QAPP	CR-REVQP	Revenue Share Premium AdjustCR	11155	5106	REVQP
QAPP	DR-DF	Developer Fee Adjustment DR	20105	5106	
QAPP	DR-ENFE	BREW Enablement Fee Adjust DR	41100	5106	ENFE
QAPP	DR-REVQ	Qualcomm Revenue Share Adj DR	41100	5106	REVQ
QAPP	DR-REVQP	Revenue Share Premium AdjustDR	41100	5106	REVQP
QAPP	IN-AMFEE	Qualcomm AnnualMaintenance Fee	41100	5106	AMFEE
QAPP	IN-AUTHE	Qualcomm Authentification Fee	41100	5106	AUTHE
QAPP	IN-CERT	Qualcomm Certification Fee	41100	5106	CERT
QAPP	IN-DF	Developer Fee	20105	5106	
QAPP	IN-ENFE	BREW Enablement Fee	41100	5106	ENFE
QAPP	IN-LICFE	Qualcomm Licensing Fee	41100	5106	LICFE
QAPP	IN-REVQ	Qualcomm Revenue Share	41100	5106	REVQ
QAPP	IN-REVQP	Qualcomm Revenue Share Premium	41100	5106	REVQP

PeopleSoft Accounts Payable

- ✓ **New Business Unit:** QAPP.
- ✓ **Gross or Net Accounting?** Gross.
- ✓ **Prorate Sales Tax:** Yes
- ✓ **Prorate Use Tax:** Yes
- ✓ **Prorate Freight:** Yes
- ✓ **Prorate VAT:** No
- ✓ **InterUnit Accounts:** 11888 & 11888
- ✓ **Journal Templates:** Standard AP templates.



- ✓ **GL Business Unit**: QAPP
- ✓ **Accounting Template:** Use current one in CORP.

More AP Setup Issues:

- ✓ **AP Payment Terms:** Use existing ones.
- ✓ **Vendor Pay Group**: None in the beginning.
- ✓ **Voucher Mask:** "QS"
- ✓ **Voucher Origin**: BRW

PeopleSoft General Ledger Setup

- ✓ QAPP has already been operating as a GL Business Unit.
- ✓ New Account: 20105
- ✓ New Products: ENFE, REVQ, REVQP, AMFEE, AUTHE, CERT, LICFE

TableSet Sharing

For Table Groups in GL and AP functions, QAPP will point to CORP SetID.
For all other functions in PeopleSoft, QAPP will point to QAPP SetID.

TableSet Sharing is set up and maintained by CORP Finance and must not be changed without mutual and prior agreement among all entities at Qualcomm.

Appendix D Build 3

1) Co-Mingled Test/Prod Environment - Provide ability to co-mingle carrier trial data and carrier production data within one Production database and provide the ability to print invoices & run reports.

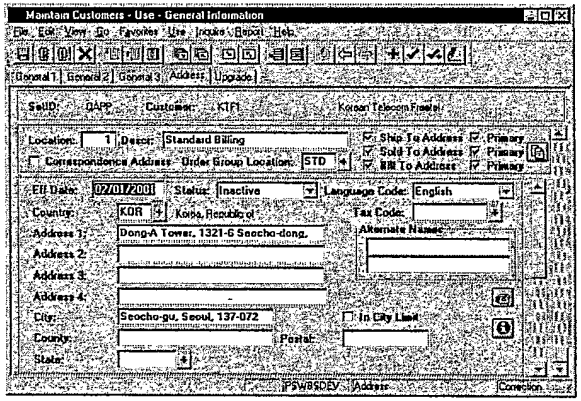
Peoplesoft

- Special Carrier setup in Peoplesoft Bill Status Change process to determine which carrier invoices to change to "Ready" and which to keep at "New" status
- A PS Query and Crystal Report to produce a Carrier Invoice regardless of status, even if it's been cancelled. This will also be used as an invoice re-print functionality for other invoices including production invoice.
- The Billing Interface SQR program (BBBI0001) has been modified to have logic that does two things:
  - ✓ Set the Carrier Header status in BREW Billing to "T" when it detects the carrier in PS to be in Test mode.
  - ✓ Nullify the AMT_TO_PAY field in MFR_SUMRY_HDR table. It will retain the original amount in the description in the MODIFIED_BY field for reference however.

Brew Billing

- Function FB_GET_ENV_MODE will determine 'Active' or 'Inactive' status from PS_CUST_ADDRESS_BB table and flag update the STATUS field in the TXN_STAGE and REV_SUMRY_HDR tables to 'Test' or Prod' for reporting.

The PS Bill Status Change process ensures that an invoice status can be changed to "Ready" ONLY IF the most current address is "Active". Otherwise, it will leave it at "New"



For a Test carrier, create a subsequent effective date address with "Inactive" status.

For a Production carrier, delete the second record or change to "Active" status.

2) Effective Premium Rate - Provide ability to add Versioning for the Manufacturer PREMIUM_RATE to record change history with effective dates for reconciliation

- Function FB_GET_MFR_PREM_RATE will lookup Premium Rate (max (version)) in the MFG_PREMIUM table for the billing period

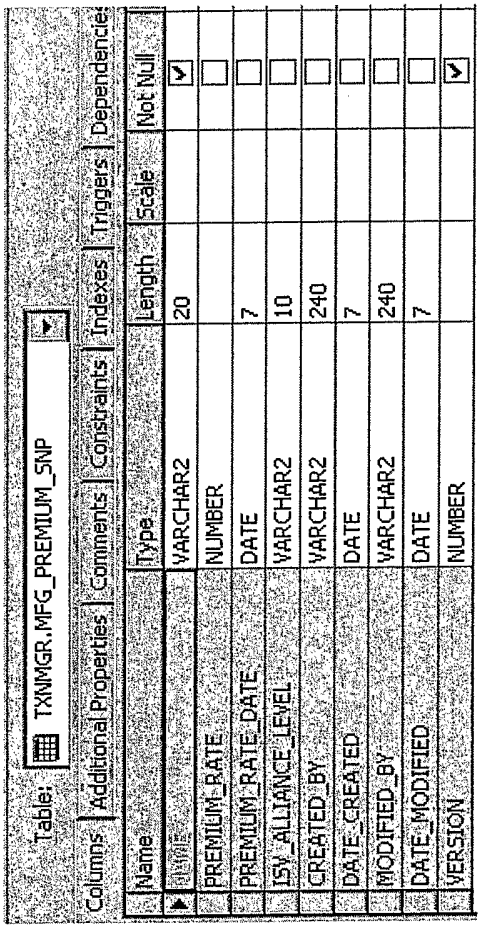


Table: TXNMGR.MFG_PREMIUM_SNP
Columns
Additional Properties
Comments
Constraints
Indexes
Triggers
Dependencies
Name
Type
Length
Scale
Not Null
PREMIUM_RATE
VARCHAR2
20
NUMBER
PREMIUM_RATE_DATE
DATE
7
ISV_ALLIANCE_LEVEL
VARCHAR2
10
CREATED_BY
VARCHAR2
240
DATE_CREATED
DATE
7
MODIFIED_BY
VARCHAR2
240
DATE_MODIFIED
DATE
7
VERSION
NUMBER

Appendix E - BREW Billing Exceptions

Brew Billing exceptions are currently logged/notified from the BREW Billing engines are categorized by the following:

- Exception Number
- Engine (Stage, Billing, Payment, Revenue)
- Exception Type
- Exception Category (Error, Oracle Error, Informational, Warning, Exception)
- Short Description
- Recipients/Email list (currently brew.billing.admin, brew.billing.acct)
- Exception Details
- Resulting Tasks

The exception list can be found in Livelink at:
http://lvcrpprd.qualcomm.com/livelink/llview/BREW_Billing_Notifications_List.html?func=doc.View&nodeId=22217010&docTitle=BREW+Billing+Notifications+List%2Edoc

91 **Table of Contents**

1 Introduction

1.1 Purpose of This Document

The QIS Distribution Center (QDC) Functional Specification defines the terminology, applications, services, data interfaces, physical characteristics and conceptual architecture for the QDC related Binary Runtime Environment for Wireless (BREW) functions. The QDC is a suite of services included in the QIS middleware. QIS middleware includes: QDC and Application Download Server (ADS) services.

The purpose of the QDC Functional Specification is the following:

- Define and clarify terminology, especially those which are used across multiple functions (i.e., Finance, Certification, Unified Application Management (UAM), Application Download, etc).
- Identify key functions and data interfaces at a high level which will serve to bridge the BREW business model with the QDC system development and test efforts.
- Serve as one of the primary document for testable functional requirements across QDC services.
- Facilitate communication and agreement relating to functional consistency across QDC teams and QIS management.
- Serve as an evolving document for specifying Phase 1 and future functional requirements.

This document primarily focuses on the Functional Requirements specific to UAM and Web Services. Refer to section 1.3 for the listing of related specifications for other QIS middleware services including Billing, Transaction Manager, CRM, ADS and Certification Centers.

1.2 Document Organization

This document contains the following sections:

- *Section 1, Introduction:* Describes the purpose of the document, document organizational structure, related documents and revision history.
- *Section 2, Conceptual Architecture:* Provides a high level functional overview of the QDC applications and services. Included in this section is the definition of the Level 0 Business Process definitions.

- *Section 3, Business Applications:* Describes the high level business applications functions included in the areas of Finance, Engineering, Logistics/Operations, and Customer Relationship Management (CRM).
- *Section 4, Unified Application Management:* Describes the UAM services in terms of catalog management, application management, ID generators, and interfaces.
- *Section 5, Middleware Tools::* Describes tools provided to external users which is needed to process applications thru the QDC middleware.
- *Section 6, Web Services:* Describes the BREW internet and extranet services.
- *Appendix A, Glossary:* Defines acronyms, and abbreviations used in the document.
- *Appendix B, Level 1 BREW Business Process Definitions:* Contains a description and diagram for the Level 1 BREW business process definitions.
- *Appendix C, Export Compliance/Encryption:* Contains a description of export compliance guidelines which is the responsibility of the ISV.

1.3 Related Documents

The following documents are related to this document and referenced herein:

- QIS Application Certification Center (ACC) CDD
- ADS Specification
- QIS Price Plan Specification
- QIS Transaction Manager (TXN) Specification
- QIS Distributed Transaction Manager (TXN) Concept Paper
- QIS Transaction Manager (TXN) Performance Benchmarking Results and Hardware Sizing
- QIS Certification Extranet Specification
- QIS Billing Support Services Specification
- QIS Billing Specification
- QIS CRM Specification
- Carrier Distribution Guide
- Developer Distribution Guide

2 Conceptual Overview

This section provides a high-level overview of the QDC conceptual architecture and the level 0 business process definitions which utilize the QDC applications and services.

2.1 Overview

2.1.1 Terminology

This section describes terms and identifiers that will be used in this document. Table 1 defines cross-functional QIS Middleware terms and identifiers. This table is not meant to be a complete list of all QIS terms but specifically those terms which are relevant to the QDC applications and services.

Table 1. QIS Middleware Terminology

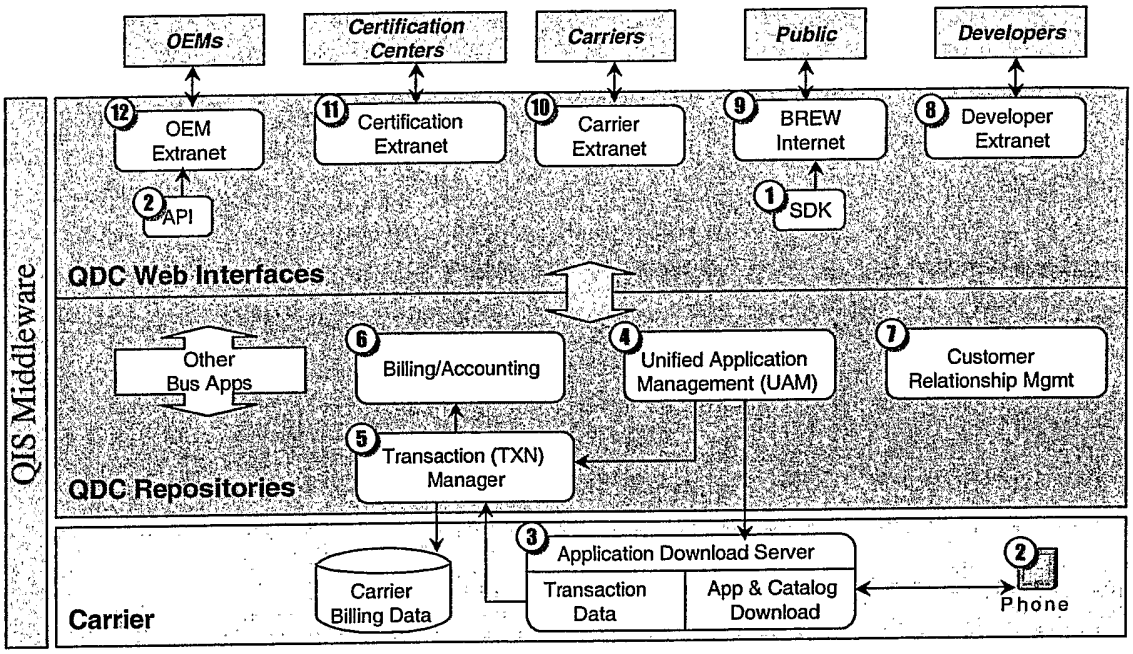
Name	Purpose
ISV	Independent Software Vendor. Term used to identify the developer company which has established a definitive agreement with QIS.
Restricted Application	A restricted application or "pass-thru" application is a non-certified BREW application submitted to QIS from the carrier. It is only available originating carrier for distribution. A carrier may be configured to either pay all ISVs of restricted applications or for QC to pay the ISVs. QIS will not support a hybrid payment model for restricted applications for a single carrier.
Limited Application	A limited application is a BREW certified application which may be set to limited distribution to a select set of carrier(s) by the ISV.
Public Application	A public application is a BREW certified application that is made available to all carriers for distribution by the ISV.
Standard Application	Either a Limited or Public Application.
Appl ID	Assigned automatically by UAM and only relevant to UAM and ADS. Part Number Relationship. UAM maintains the mapping between part numbers and Appl IDs. Transaction History: Appl ID is contained in the transaction history and used to track phone events.
SID	Subscriber ID. This is a pseudo customer identifier that is used to identify customers. It may correspond to a MIN (i.e., phone number) or may be another carrier designated identifier. The SID and MIN will be logged at the ADS with the events will be provisioned on the phone. If a SID is provisioned on the phone, MIN will not be transferred back to TX.
MIN	Mobile ID Number or phone number The MIN is used for transaction processing if the Carrier elects not to provision a SID. In that case: Transaction History: MIN is passed in the transaction log for billing and payment processing. Provisioning Data: Carriers will provide a perioding report to QC which will include MIN updates (i.e., MIN out of service, MIN reassignment, etc.)
Part Number	▪ Is assigned by ACCHQ when the application is submitted for certification. ▪ For Restricted Applications, the Carrier assigns a carrier unique part number and then QDC Operations adds a carrier prefix to the carrier part number. ▪ Part numbers are used for tracking the distribution and of BREW applications. It is a number exposed externally and a key identifier for billing process functions (i.e., not Appl IDs)
Platform ID	A platform is a single or set of phone type(s) by manufacturer, model, and s/w revision that are compatible in hosting BREW applications. The platform ID is the identifier for a platform. This is

	used by ADS and UAM for deriving the phone display catalog. A platform ID can be a superset of platform IDs
Pkg ID	An application can be composed of multiple packages. Once package is associate to one module. UAM generates that Pkg ID and the association of packages to module ID.
Module ID	The MIF file contains list of classes. When processing the application into UAM, one module ID is associated with a list of classes contained in the MIF file. A module is associated to multiple packages. A module id is a key identifier of an application. It is thru the module ID that the application tool determines a new application or an upgraded application. Module ID is used as the directory structure naming convention in BREW.
Class ID	Similar to a dll, these are the binary object namespace and may only be exposed in a single home directory represented by the module id. The Class ID needs to be unique across all BREW applications. The Class ID is needed at compile time and is also stored in the .MIF, which is submitted by the developer.
Applet ID	Applet IDs are Class IDs, but are classified differently so the phone knows which classes are applications or support libraries. The Applet ID is needed at compile time and is also stored in the .MIF, which is submitted by the developer.
Primary Carrier	Associated to the carrier with which all BREW agreements are negotiated and approved. A primary carrier may be associated with "affiliate carriers" (i.e., for regional support).
DAP	Developer application price. This is the price which is used to determine the payment to issue the developer based on phone transactions. A developer may have different application DAP's across different carriers. It is independent of carrier's purchase price.
Purchase Price	The amount that the carrier will charge the device user.
Developer Fee (DF)	The amount that the developer is paid for the application.
QC Revenue Share (RS)	The amount that QC will charge the carriers based on application sales.
Pricing Methods	There are five pricing methods: 1) Demo (includes Trial applications) 2) Purchase (includes one-time download and unlimited values) 3) Subscription 4) Upgrade (i.e., patch is a free upgrade) 5) Provisioned (must be authorized by carrier)
Price Basis	There are four basis types associated with application pricing. 1) Fixed Uses. Billing based on 5 plays, 10 plays or unlimited. Developer/application is responsible for decrementing the counter. 2) Fixed Date. Billing based on expiration date ie: 3/31/00, date expressed as seconds since 0:0:0 6-Jan-1980 or unlimited 3) Fixed Duration. Billing based # of days after activation i.e., 30 days from first use or unlimited 4) Elapsed Time. Billing based on minutes of use ie: 120 minutes or unlimited
Price Point	For each price basis, one or more price points may be defined depending on the price method. The price point includes a value and a DAP, and the purchase price. For example, for fixed uses a price point could be 5 uses, $1.00 DAP, and $1.50 purchase price.
Price Plan	The price plan is a term used to refer to the entire pricing structure for a particular application/part number. The price plan includes all relevant price methods, basis types, and price points. It does not include Purchase Price.

2.1.2 Conceptual Architecture

The QDC provides applications and services relating to the QIS middleware as depicted in Figure 1.

Figure 1. QIS Middleware Conceptual Architecture



These QDC applications and services include the following functions:

OEM Extranet (12) – The OEMs will be provided extranet services to assist them in provisioning the wireless devices with the BREW API, MobileShop, BREW applications and other BREW required components. The OEM extranet will also enable the OEMs to request the creation of additional BREW platform IDs and to submit Restricted Applications for signing and packaging

Certification Extranet (11) – The Certification extranet provides services which facilitate communication between the BREW certification centers (initially NSTL) and the Certification Center headquarters in San Diego. Services provided via the extranet include: a) ESN management functions for generating test signatures for applications to be executable on test phones; b) catalog management services for managing the certification center catalogs; c) certification status management for tracking whether an application's certification status is submitted, certified or failed; d) access to certification metric reports; and e) access to applications and documentation.

Carrier Extranet (10, 4) – The carrier uses the carrier extranet application service to manage the carrier catalog which contains the selected BREW applications to distribute to their consumers. Applications selected by the carrier for distribution will be replicated to the Carrier ADS (3) along with the carrier catalog updates. The phone (2) will interact with the Carrier ADS (3) to download the application to the phone and activate the application for the phone user. Phone transactions are logged at the ADS and uploaded to the transaction manager for Billing conversion and rating (3,5). In

addition, the carrier extranet provides billing support services that augments the invoicing and financial reconciliation process with QIS and carriers. The carrier extranet will also contain facilities for submitting and processing of Restricted Applications from the carrier.

BREW Internet (9) – The BREW internet pages will be available to the public off of the QC internet site. The pages are targeted to provide general BREW information to the public, developers, OEMs, and carriers. General BREW marketing information, white papers, technical information etc, will be available on the internet. In addition, developers will be able to download the BREW SDK and access to basic technical information relating to BREW and the SDK.

Developer Extranet (8) – The Developer Extranet provides BREW services to the ISV. Included in these services is the ability for the ISV to maintain the application price plans and access to ISV billing support services. In addition, there is access to various BREW development tools, documents (i.e., Carrier Guidelines), access to certification center, and the procedures for a developer company to sign on as a BREW ISV.

Unified Application Management (UAM) (4) – UAM is a repository which will manage certified and pre-certified applications as well as carrier catalogs. Once an application is certified by the Certification Center, the ACCHQ sets the application "ready for distribution" to the UAM via BREW Admin. In addition, Restricted Applications enter the UAM after going through ACCHQ only for signing. Once the application is marked ready for distribution the application is made available for carrier distribution via UAM (4) and the Carrier Application Catalog Management functions on the Carrier Extranet (10).

Transaction Manager (TXN) (5) – Transaction Manager is a data repository which receives uploads of phone transaction data from multiple ADS farms. TXN consolidates the raw phone transactions and then processes the data by converting information and applying pricing to each transaction (i.e., a rating process). Once the transactions are converted and rated, Billing and Accounting programs process the information and prepare to process financial transactions (6). These financial transactions include carrier invoicing and developer payment. TXN is also used as the primary repository for applying adjustments and generating Carrier Billing Extract files to enable the carrier to bill their phone users for application purchases.

Customer Relationship Management (CRM) Services (6) – Carriers, OEMs, Developers, Customer Prospects and Certification Centers will be provided customer support services. Customer tickets will be managed via the CRM system and an escalation path will be implemented for addressing customer support requests.

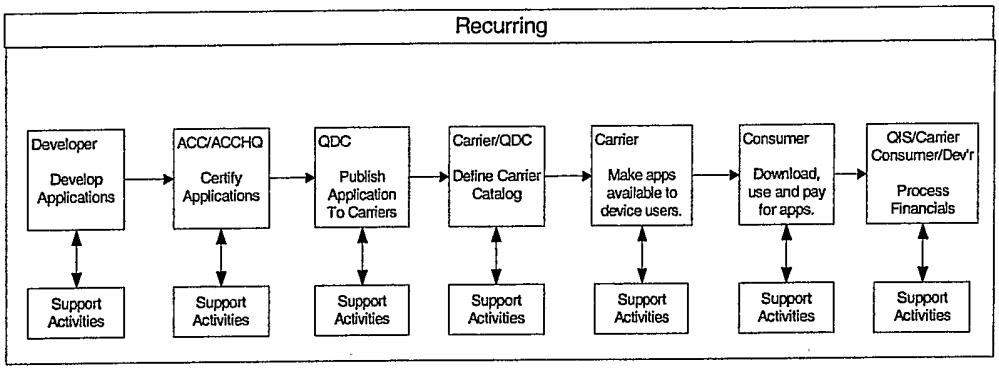
Billing/Accounting Services (6) – As part of the distribution services QC will provide accounting "clearing house" services for the primary carriers and developers. As such, the QDC will collect phone transaction information which will be used in conjunction with carrier provisioning data and accounting adjustments to compile a set of transactions as the foundation for invoicing the primary carrier. This invoice shall include QC fees and developer fees based on application usage and according to the carrier agreements. Refer to the QIS Billing specification for more details relating to carrier invoicing and developer payment.

2.2 High Level Business Process Definitions

2.2.1 Level 0 Business Process Definitions

Figure 2 illustrates the Level 0 business processes that comprise the QIS/BREW business model. It is a simplified process flow which starts with the developer creating an application using the BREW SDK through the certification process and distribution of the application with the carriers. Finally, the developer, carrier, QC and consumer are billed and collect payments accordingly. This Level 0 process definition provides the high level framework for Level 1 detailed process definitions.

Figure 2. Level 0 BREW Business Process



2.2.2 Level 1 Business Process Definitions

Each process within the Level 0 diagram is further defined and illustrated in the following Level 1 process definition diagrams. Each diagram details the process from a different perspective (i.e., developer, carrier, consumer, etc.). For each perspective, the business process definition identifies functions across distinct phases of the process. For example, the developer is in pre-certification, when he/she has downloaded the SDK and is developing an application for application certification/distribution. When the developer submits the application to an Application Certification Center (ACC) for certification he/she is entering the certification phase. At this point the developer must be associated with an ISV to enter certification phase. Once the application is certified by the ACC the ISV enters the post-certification phase (i.e., distribution phase). The process definitions are organized according the process phases identified in Table 2. The process definition diagrams can be found in Appendix E of this document.

Table 2. Level 1 Process Definition Diagrams

Level 1 Perspective:	Process Phases:	Document Location
Application Certification	Developer Authentication Application Submittal Application Management & Testing Notifications Pass Through Applications	Appendix B, Section 1.0
Developer	App Development & Submittal Application Certification Set Pricing and Distribute Receive Payment	Appendix B, Section 2.0
Carrier	Negotiate and Select Apps	Appendix B, Section 3.0

	Submit Restricted Applications App Availability / User Trials Production Availability Accounting and Collections	
Consumer	User Invoked Activity Download (transact) apps Use & Pay for apps	Appendix B, Section 4.0
Distribution Center	Receive certified apps Build / Maintain Carrier Extranet Manage Apps via UAM Support Handset Manufacturers Transact Financial Data	Document in development
Handset Manufacturer (OEM)	Submit Restricted Applications Setup Platform IDs Receive Preinstall Applications	Document in development

403

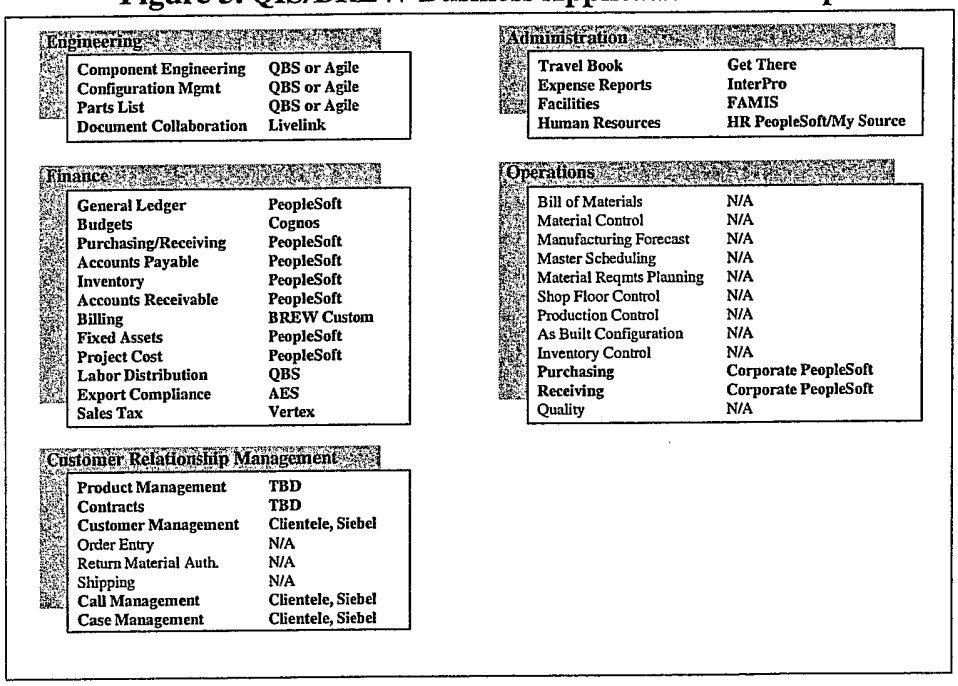
404

3 Business Applications

3.1 Business Applications Roadmap

Business Applications include Accounting, Engineering, Customer Management, Administration and Operations services. Figure 3 depicts the current Phase 1 QIS/BREW business application roadmap. In Phase 1, the focus of efforts will be in the custom billing functions needed in Finance and the CRM services. BREW will utilize existing business application services for the remaining business system functions.

Figure 3. QIS/BREW Business Application Roadmap



3.1 Finance

3.1.1 BREW Finance Overview

This section provides an overview of BREW Finance functions, which are being customized to support Carrier Invoicing and ISV Payment. Refer to the QIS Transaction Manager Specification and QIS Billing Specification for further detailed requirements.

3.1.2 Transaction Data Flow

Figure 4 illustrates the more detailed data flow of transaction and billing data.

Step 1: Application Download Server (ADS) Transactions

Application Download Server collects the phone transaction data and uploads the transaction data to TXN periodically (i.e., every 30 minutes)

Step 2: Transaction Manager (TXN)

The transaction manager consolidates the transactions across all ADS farms. TXN then processes the transaction data converting the IDs (i.e., App IDs to part numbers) to billing relevant IDs and filtering out non-billing event data (i.e., debug or monitoring events). In addition TXN will derive the developer DAP for each transaction so that billing has complete and denormalized set of billing event data for subsequent processing. The billing event data is used as the "data of record" for processing billing related transactions and for generation of the carrier billing extract file(s).

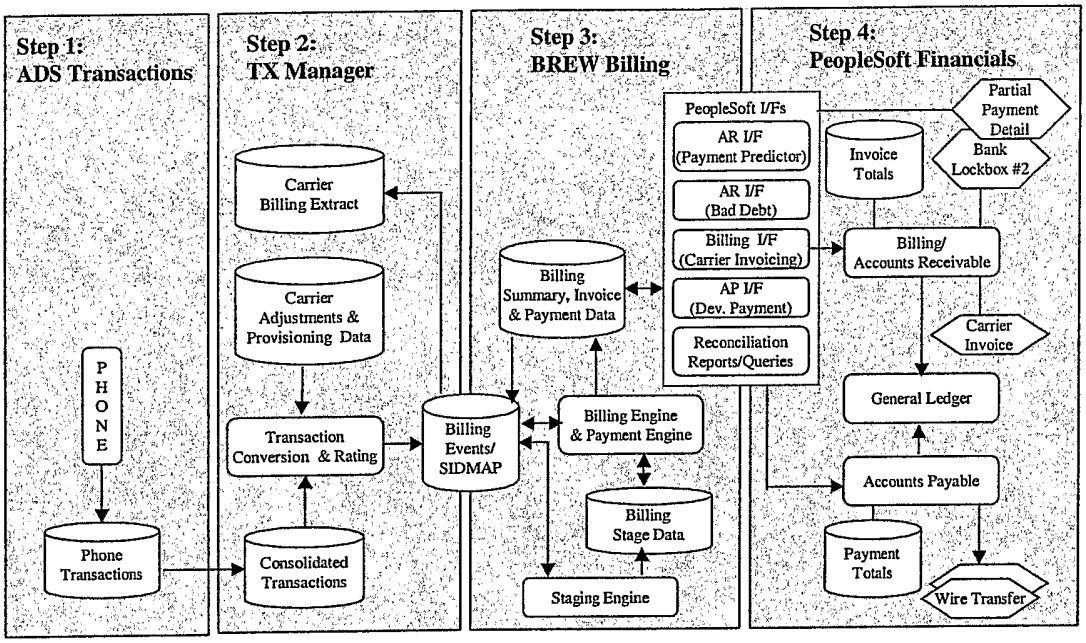
Step 3: BREW Billing (BB)

BREW Billing is responsible for deriving the ISV payment for Standard Applications and carrier invoice summary and totals from the detailed billing event data. Because of the volume of transactions to process (i.e., 6 million per day), BREW Billing will subtotal and "stage" the billing data on an incremental basis. This alleviates processing and entire carrier monthly period of billing in one job. BREW stage will process incremental subtotals of the billing event data and store these subtotals in billing stage. The billing invoice and payment engines will subsequently processing the billing stage data and derive billing summaries and totals for the billing period which will be stored in the billing summary data. Once the summary data is calculated, PeopleSoft interface programs will run which processes the summary data, computes the invoice and payment grand totals and invokes the Billing/Account Receivable and Accounts Payable PeopleSoft functions. Taxes are processed thru PeopleSoft.

Step 4: PeopleSoft Financials

PeopleSoft will be used to process the carrier invoices and ISV payment for billing totals. The billing summary and detail will be retained in TXN and BREW Billing. In addition, PeopleSoft AR will provide the cash application functions. An interface from PeopleSoft AR to BREW Billing will be developed which tracks payment back to the billing summaries and subsequently to the billing event data. This is required in order to drive the ISV payment appropriately (i.e.,In the basic plan, the developer gets paid only when the carrier pays QC).

Figure 4. Transaction/Billing Data Flow



3.2 Engineering CM

Any release software and documentation that is used outside of QUALCOMM San Diego is required to go through Engineering configuration management (CM) processes.

In Phase 1, this requirement would include:

- ADS: Includes CM of all BREW carrier site software and documentation.
- QDC: User guides and documentation.
- ACCHQ: Includes any Qualcomm developed application certification tools, testing procedures, selection and start up criteria or audit procedures.
- BREW: Phone software (MobileShop and APIs) and Porting Kit

In future phases, when QDC distributes any of the QDC software (Transaction Manager) outside of QUALCOMM San Diego, the software and supporting documents will have to go through normal QUALCOMM CM processes.

3.3 Logistics/Operations

3.3.1 Application with QC Part Numbers

a) Applications will be assigned QC part numbers or Carrier part numbers. Specifically, restricted applications will be assigned a Carrier part number. Limited and public applications will be assigned QC part numbers.

b) The QC part number is a sequential number and generated by the UAM. The QC part number will be formatted to "QC" plus 8 numeric (i.e., QC00000001).

c) For restricted applications, carriers will provide BREW with a carrier specified part number. This part number may be 40 alphanumeric.

d) ACCHQ personnel will associate and assign QC part number and part name to certified/restricted applications for tracking purposes.

e) The relationship or association between the part number and the application is stored in UAM.

f) Multiple application IDs may map to a single part number. There is only one part name per part number. It is the part name that is exposed externally for billing purposes which includes carrier billing extract and in billing functions (i.e., carrier invoices and developer payment).. Application name is displayed in the carrier catalogs.

g) The QC Part number will not be generated through QBS (ie: they will not be MCNs) or tracked via Qualcomm's CM function.

3.3.1.1 Changes to Part Numbers

a) A part number will be changed when:

- Form, Fit or Function Change (FFF)
- Priced upgrade to an Application

b) A part number may change when:

- New supported language – this is a business decision
- Open Issue: Different Price Plan for Different Handsets/Platform IDs

c) A part number does not need to change when:

- DAP changes
- Application patch is released
- Carrier purchase price changes
- Developer modifies pricing method or basis attributes
- Open Issue: Carrier certifies application independently

d) Part Number Relationship to Application Languages

- Applications, which support different languages, may or may not map to the same part number.

- Depending upon carrier agreements, QC may require that applications of different languages be assigned different part numbers for billing purposes.
- If unique part numbers are not needed for exercising the carrier agreements, the part numbering strategy is a business decision. It may not be required to be implemented consistently across applications.
- UAM shall have the flexibility to accommodate the same part number and different part number scenarios for applications in different languages.

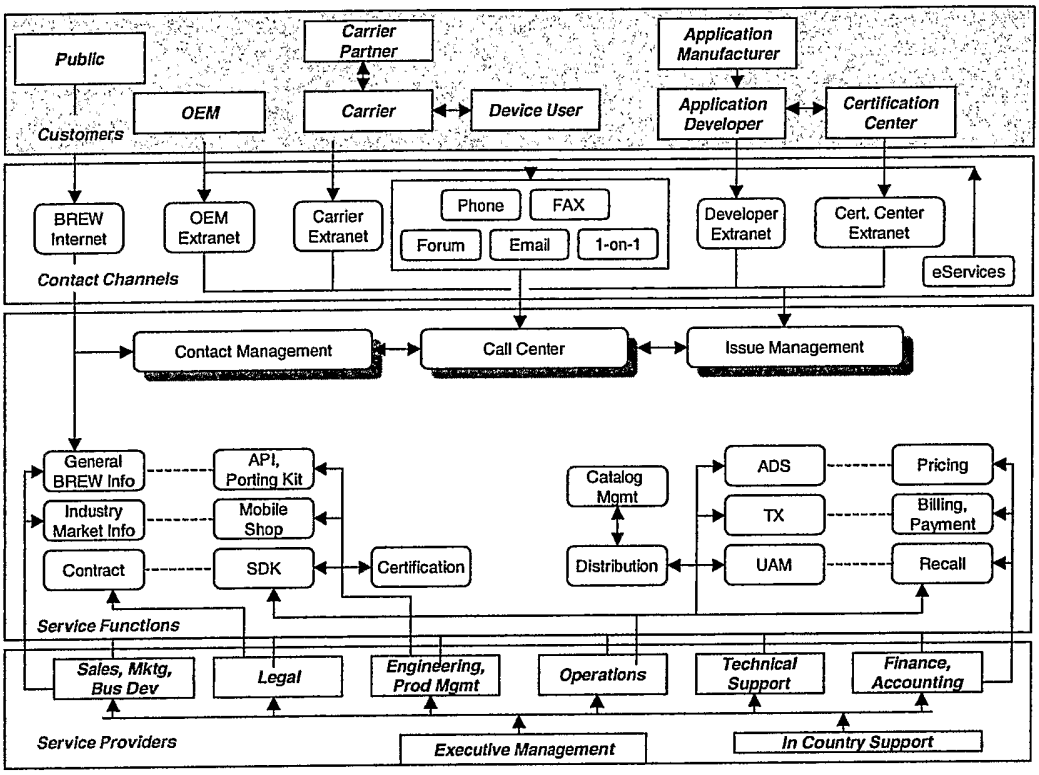
3.4 Customer Relationship Management (CRM)

This section introduces the various QIS customer relationship components and provides a brief description of each functional group. Additionally, high-level business processes of the QIS value chain are identified. Refer to the QIS CRM Specification for further detailed requirements.

3.4.1 CRM Conceptual Overview

Figure 5 provides a conceptual overview of the current QIS customer relationship components. The overview is categorized into four key areas: Customers, Contact Channels, Service Functions and Service Providers. Following sections explore each of these areas in detail.

522 **Figure 5. QIS Customer Relationship Management Overview**



523
524

525 3.4.1.1 Customers

526 QIS's BREW™ customers will fit into one of the six categories below:
527
528 *Public*: The BREW™ Public is made up of the global user community who may have
529 learned about BREW™ from any number of the sources including: direct mail,
530 advertising campaigns, promotional material, colleagues, published articles,
531 Carriers, and the Internet. Once the "Public" customers learn more about
532 BREW™ they may parlay themselves into another BREW™ customer category,
533 such as a Device User or Application Developer.
534
535 *Original Equipment Manufacturer (OEM)*: A BREW™ OEM provides the devices
536 (phones, PDAs, etc.) for the Public and Device Users. BREW™ applications will
537 run on these devices.
538
539 *Carrier*: A BREW™ Carrier provides the channels for selling the BREW™
540 applications. These applications will be developed by Application Developers for
541 consumption by the Carriers' customers.
542
543 *Device User*: A BREW™ Device User is an end user who has a BREW™ OEM
544 wireless device and subscribes to a BREW™ Carrier's wireless service. Device

Users can be located anywhere in the world. They use wireless applications developed on BREW™ and provided by BREW™ Carriers.

Application Manufacturer: A BREW™ Application Manufacturer is a developer house with many Application Developers. Application Manufacturers develop applications using the BREW™ Software Development Kit (SDK) and submit them for certification. The Application Manufacturer community will be spread across the world. Following certification of an application, a BREW™ Carrier may offer it to their customers. The Application Manufacturers have the option to submit them to a Certification Center for certification or submit them directly to a Carrier without certification.

Application Developer: A BREW™ Application Developer develops applications using the BREW™ Software Development Kit (SDK) and submits these applications for Certification. Such developers may or may not belong to an Application Manufacturer. The Application Developer community will be spread across the world. They follow a certification process similar to the Application Manufacturers.

Certification Center: A BREW™ Certification Center tests the Application Developers' applications for stability, and compliance per QUALCOMM (QC) and Carrier's guidelines. These centers can be located worldwide and will maintain certain amount of independence from QC. Once an application is certified, it is termed as "True BREW™" and is ready for a Carrier to provide to its customers. Carriers may provide some "Pass Through" applications to their customers, which are not certified by Certification Centers and are termed not "True BREW™". Application Certification Centers' activities will be coordinated through a Qualcomm owned Application Certification Centers Head Quarters (ACCHQ)

3.4.1.2 Contact Channels

QIS will utilize every available channel to promote BREW™ and serve its users.

Internet (brew.qualcomm.com): Currently the BREW™ Internet site serves mainly as a sales and marketing tool, and is geared to provide basic information about BREW™. It allows visitors to sign-up and receives additional information about BREW™. The site's functionality is expected to expand in the coming months.

Extranets (OEM, Carrier, Application Developer, Certification Center): The BREW™ suite of Extranets has three main purposes:

- A locale where more detailed, customer specific BREW™ information can be obtained, such as pricing templates, certification testing process definitions, a listing of applications available for hosting, and ID generators.

- A proactive tool for addressing possible customer questions and issues that may impede BREW™ implementation. Tools include Frequently Asked Questions (FAQs), a knowledge base with known issues and solutions, and an issue submittal area.

- A secure, customer specific area, where confidential information can be exchanged openly between QUALCOMM and a customer.

Phone, Fax, Email, On-Line Forum, and Face-to-Face Meetings: QIS will staff the necessary resources to receive and respond to phone, fax, email, and web inquiries. Additionally, QIS staffers will attend CDMA conferences to promote BREW™, extend understanding, and address any questions.

eServices: QIS will provide customers with a variety of web enabled self-service tools. These tools include: interactive voice response systems, auto e-mail responses based on key word searches, auto e-mail confirmation, knowledge bases with problems and known resolutions, on-line chat sessions, on-line technical forums where a developer could post a question and seek answers from others in the BREW™ community, usage reports etc. These self-service options, besides savings cost, will be available 24/7 to QIS customers, and minimize the burden on the QIS technical support representatives. The QIS Extranets will be used to facilitate some of these functions.

3.4.1.3 Service Functions

The Service Functions can be grouped as follows:

Systems: In order to capture interactions with customers and products, QIS will provide the following services:

- Contact Management: a central repository for all contact information pertaining to a customer, prospects, etc.
- Call Center: a central repository of all customer calls, their resolution, and call history.
- Issue Management: a central repository for all issues encompassing product, process, customer etc. and tracks their assignment, resolution, and history.
- Opportunity Management: a repository of all business opportunities as identified by any authorized QIS service providers

Functions: QIS service providers (identified below) will support the BREW™ customers in the following areas: General Information, Contract, API Porting Kit, Mobile Shop, SDK, Certification, Extranet, Transactions (TX), Unified Application Management (UAM), Application Download Server (ADS), Catalog Configuration,

3.4.1.4 Service Providers

Following are QIS service providers and a list of their services to support BREW™. All QIS departments will have dedicated skilled resources to service escalated customer inquiries. Escalated customer inquiries will be sent to the appropriate QIS department for resolution.

Marketing –

- Promote BREW™ and serve as the first line of contact for external world
- Qualify prospects and forward them to the appropriate QC service providers
- Attend conferences and other promotional events
- Generate press releases, announcements etc.

Business Development –

- Promote BREW™ in concert with Marketing group and generate new business leads
- Interact with and evaluate developers, business partners, Carriers, OEMs, suppliers, and upon qualification forward them to the appropriate BREW™ service providers.

Sales –

- Negotiate contracts with Carriers
- Oversight and Carrier champion during implementation
- Promote BREW™ through all possible avenues
- Facilitate matching of carriers and application developers for popular applications
- On-going liaison with Carriers

Legal – The Legal team supports Application Developers, Carriers, OEMs, Suppliers in developing Memorandum of Understanding (MOU), BREW™ related contracts.

Engineering – The Engineering team's support will be split into: Development Engineering, Support Engineering, and Test Engineering. These teams will support BREW™ in several ways:

- Provide support to Application Developers for API Porting Kit, Mobile Shop, and SDK.
- Application Certification
- Support application recall process,
- Facilitate BREW™ user and carrier trials

Product Management – The Product Management team shall:

- Receive feedback and enhancement requests
- Work with Engineering and Business Systems to address the requests and develop a release strategy
- Promote BREW™ through all possible avenues
- Develop new concepts and/or requirements for BREW™
- Develop the next product release and/or product upgrades
- Verify that developers' contracts are in order.

Business Systems – The Business Systems team support spans across various BREW™ functions: Recall, ADS, TX, UAM, Catalog Management, Distribution, and other Extranet functions.

- Develop and maintain QIS middleware
- Create and maintain Extranets
- Build and maintain the QIS Data Center
- IT Coordination
- Regular monitoring of the ADS servers to quickly address operational issues such as slow downloads, response times
- 24/7 support during application recall process until the situation is resolved

- Publish general system health information on a web page for general availability within QC

Technical Support – QIS Technical Support service providers will be the front-line and interface with all QIS customers. They provide:

- In-country support through local language speaking staff during business hours (at a minimum)
- US based customer support during business hours
- Support the application recall process
- Escalate customer's inquiry to appropriate QIS Service Provider as needed

Product Support –

- Provide Tier 1 support for developers and Carriers for pricing, billing, payment issues. If issues cannot be resolved, escalate them to appropriate QIS Service Provider
- Facilitate resolution of billings' issues and billing related inquiries by working with the Carrier Account Manager
- Will associate and assign QC part number and part name to certified/restricted applications for tracking purposes
- Verify validity of Application Developers information coming from web page "Become a BREW™ Developer".
- Add vendor approval process to QC for verified Application Developers
- Check with Legal and Product Management if hard copy of Application Developer agreement with real signature is needed vs. automated agreement. If human intervention needed, support Application Developer questions regarding status of legal agreements in coordination with Contract Administrator.
- Continue coordinating FUT's

Finance –

- Negotiate agreements
- Pricing and other financial aspects for Application Developers, Carriers, Certification Centers etc.
- Provide Tiered support for financial matters
- Work with the Carrier Account Manager to resolve any financial issues

Accounting –

- Handle pricing, billings, and payments to Carriers and Developers
- Provide information to extranets for Application Developers and Carriers with information related to billings, payment status, check date, adjustments, bad debts etc.
- Responsible for billings and collections for Certification Centers
- Billing adjustments during recalls

- Provide support for issues related to bill delivery to Carriers and payments to Application Developers
- Support export compliance and tax issue resolution
- Work with the Carrier Account Manager to resolve any financial issues

In-country Support --

In addition to all the Service Providers identified above, QIS will offer in-country support to its customers. The extent of this support, delivery mechanisms, and service providers is yet to be finalized. Currently the following type of support is being discussed:

- **Call Center:** QIS may provide in-country local language speaking technical support to its customers including – developers, OEMs, carriers, suppliers, and certification centers. QIS intends to outsource this service to local firms, however, a decision is not yet reached. Once the outsourcing process is finalized, QIS will establish processes for tiered support, data interfaces, escalation procedures etc.
- **Certification Centers:** Third party Certification Centers are expected to be located in US, Asia Pacific and other global locations. These entities will receive certification guidelines from QIS and will be completely responsible for the certification process, including handling Application Manufacturers and Application Developers inquiries. QIS expects to partner with these firms.
- **Carriers:** Carriers will offer their own in-country local language speaking tiered support services to device users. It is expected that Carriers will handle a majority of the device users' inquiries, thus only a fraction of these inquiries will be escalated to QIS. QIS expects to establish an interface mechanism with these firms.

4 Unified Application Management (UAM)

UAM is the centralized suite of services for application and distribution management. These services include:

- Carrier Catalog Management : enables the carriers to manage the organization and distribution of applications to their carrier ADS.
- Application Management: includes the version control of the application files and associated metadata.
- Document Storage: includes a file system structure for storage of application documents (i.e., test documents, user guides, etc.)
- ID Management: generates various IDs and manages metadata associated with IDs and their relationship to other UAM objects.
- UAM Interfaces: supported interfaces relating to UAM data stores and needed by other BREW middleware services (i.e., ADS services, web applications, billing services, etc.)
- UAM E-Mail Notifications: generates e-mail notifications, as required to carriers, developers, and QIS personnel.

Functional requirements for UAM services are described in this section.

4.1 Carrier Catalog Management

The carrier extranet will enable authorized carrier users to manage their carrier catalogs. The carrier extranet will be implemented using UAM APIs. These APIs will abstract the application from the implementation specifics of the UAM data dictionary. The following sections will identify the key catalog management functions that UAM will be designed to support. Refer to Section 6 for further identification of carrier extranet catalog requirements, a majority of which are directly related to UAM repository services.

4.1.1 BREW Master

a) The BREW master includes all applications and associated metadata managed by UAM.

b) The BREW master consists of all applications (i.e., certified, pre-certified and non-certified/pass-thru) managed by UAM.

c) The BREW master will contain applications available for "public, limited and restricted" carrier distribution.

d) The BREW master is only administered by authorized QDC personnel, because it contains carrier sensitive information.

4.1.2 Carrier Master Applications List

The carrier master applications list is a system derived list of carrier applications that applies all carrier restrictions and limited distribution rules. The carrier master applications list is used to apply carrier global restrictions.

4.1.3 Carrier Global Restrictions and the Carrier Restricted View

The carrier restricted view is another derived list which takes the carrier master applications list and applies the following restrictions. They include:

a) Exclude applications by language(s)

b) Exclude applications by ISV(s)

c) Exclude applications by platform(s)

4.1.3.1 Carrier Restricted View and Active Carrier Catalogs

The carrier restricted view is a "saved filter" for the carrier administrator to use as an administration convenience tool. If the administrator changes the global restrictions, which then changes the carrier restricted view, there is no relationship between applications which have been added or removed from the carrier restricted view and applications in the active carrier catalogs. There may be an application in an active carrier catalog that is not visible in the carrier restricted view because global restrictions were subsequently updated.

4.1.4 Carrier Parts List

The carrier parts list is a created and maintained list for each primary carrier. It represents all applications with price plans "selected" by the primary carrier and available for distribution via carrier catalog management services. All applications in the carrier parts list may not be in an active carrier catalog. The carrier parts list is the "pick list" for adding applications into the carrier catalogs.

4.1.4.1 Dot Releases

There may be multiple versions of an application displayed per part number in the carrier parts list (i.e, for patches).

4.1.4.2 Purchase Price Modifications

The carrier administrator can configure whether a part number purchase price can be modified by catalog administrators or not. If not, a part number will be sold for the same price across all carrier catalogs.

Open Issue: For Phase 1, the list price must be set manually by the carrier administrator in each catalog. Future challenge when a carrier may have catalogs with different currencies for list price.

4.1.5 ADS Types

The following types are associated with an ADS:

a) Certification. This is a pre-production ADS used by certification centers to test applications during the BREW certification process.

b) Test. This is a pre-production ADS used to test applications in a pre-release mode (i.e., Carrier user trials).

c) Production. This is the production system used for commercial operations. All applications and catalogs posted the production ADS are intended to be available for commercial release.

838 ### *4.1.6 Catalog Status*

839 Catalogs may exist in one of the following states:

840 a) PENDING – a new/draft catalog that is being edited. A catalog in PENDING
841 status may be edited, cloned and deleted.

842 b) READY – a catalog, where editing is complete and which is available to be
843 pushed to an ADS. A catalog in READY status may only be cloned, set back to
844 PENDING for editing. It may not be edited in the READY state.

845 c) ACTIVE – a catalog which is currently activated on an ADS. A catalog in
846 ACTIVE status may only be cloned. It may not be deleted or edited.

847 d) DEACTIVE – a catalog which is no longer ACTIVE, probably superceded by
848 another catalog push to the ADS. A catalog in DEACTIVE status may only be
849 cloned. It may not be deleted or edited.

850 The catalog administrator shall have the ability to view the list of catalog versions by
851 state and ADS.

852 When a catalog is superceded by a be new catalog on an ADS it may not become
853 DEACTIVE because it may still be ACTIVE on a difference ADS (i.e., a single catalog
854 may be used across multiple ADS')

855 When a catalog is moved from READY back to PENDING, UAM must clean-up all
856 children records for the catalog (i.e., XML data).

857 ### *4.1.7 ADS to Catalog Association*

858 a) There is an association of a carrier's catalog/version to a designated application
859 download server.

860 b) Each ADS farm can host multiple ADS servers but only one catalog version.

861 c) An ADS will have an associated name. By convention and for administration
862 clarity, the ADS name in the catalog should be kept manually consistent with the
863 name of the ADS host.

864 d) A primary carrier may be associated with multiple ADS farms.

865 e) If a primary carrier has multiple ADS farms, the carrier may assign a specific
866 catalog version to a designated ADS farm, and/or assign the same catalog version
867 to multiple carrier ADS farms (i.e., same regional catalog across all regional
868 carrier ADS farms)

869 f) The phones will be provisioned with a set of authorized ADS farms. The
870 consumer will navigate and select the specific ADS to connect to via the phone
871 interface.

872 ### *4.1.8 Catalog Versions*

873 a) UAM will provide for versioning of the catalogs. An authorized user may create
874 a new version of a catalog or view/modify an existing version.

875 b) There is only one active catalog version on a single ADS farm at any point in
876 time.

877 c) Only one user can edit a specific version of the catalog at any point in time.

4.1.9 Catalog Attributes

4.1.9.1 Catalog Effective Date/Time

a) Once edits to a particular catalog version have been completed the catalog is set to READY status by the Catalog administrator, the ADS administrator associates the catalog version to a target ADS farm for propagation of the catalog on a specified effective date/time.

b) UAM shall capture the audit trail by user of catalog versions propagated to ADS farms and the effective date/time.

c) The effective date/time shall be capable to be specified to the day/hour/minute in local time. Note: for Phase 1 a carrier may have one offset timezone define. In the future a carrier may define multiple timezone (i.e., by sub-carrier or ADS Farm, this is TBD).

4.1.9.2 Catalog Currency

a) One currency is selected for each catalog. In Phase 1, a carrier is limited to one currency across all carrier catalogs.

b) The purchase price will be displayed in the catalog currency.

c) A carrier may have catalogs of different currencies (Post Phase 1)

d) The DAP will be managed in U.S. currency.

e) As regional carriers are implemented, the currency strategy will need to be refined.

4.1.10 Catalog Rollbacks

UAM shall support the ability to rollback the version of a catalog to a previous active version (i.e., to back-out changes) on an ADS. Rollbacks apply to all ADS types. For Phase 1 the Catalog Rollback will be implemented through cloning the previous catalog which was deactivated and activating the cloned version of the catalog.

4.1.11 Categories

a) A catalog will be organized by categories for consumer display.

b) The Catalog administrator will be permitted to create, rename, order and delete categories within a catalog.

c) For Phase 1, A catalog can have one flat list of categories (i.e., non-hierarchical categories).

d) In the future, a catalog will have the ability to contain hierarchical categories.

e) The carrier should be able to associate icons to categories within a catalog.

4.1.12 Application List within a Category

a) Each category is associated with an application list.

914 b) The Catalog administrator will be permitted to add, order and delete applications
915 within a category.

916 c) The carrier will be permitted to associate icons with applications in a catalog.

917 4.1.12.1 Application Name and Part Name

918 a) The application name displayed in the catalog is the application name associated
919 with the application/version.

920 b) The part number and part name will be used for all correspondences with ISVs
921 and Carriers (i.e., carrier billing extract files, carrier invoices, billing reports,
922 ISV AP vouchers, etc.)

923 c) The part name will be kept consistent with the application name assuming that
924 all application IDs mapping to a single part number are the same. If not, a single
925 name will be assigned to the part number by the ACCHQ. Refer to section 3.3
926 for further details on application to part number mapping and part naming..

927 4.1.12.2 Append Characters to Part Name (Post-Phase 1)

928 a) The carrier will be permitted to append characters to the application name which
929 is displayed in the carrier catalog.

930 b) This may be useful to tag applications with a visible identifier (i.e., new, bargain
931 deals, etc.).

932 c) The intent of the append function is not to disassociate the application with the
933 original name but to enable the application to be distinguished through special
934 characters.

935 ### *4.1.13 Application Patches*

936 a) A "patch" is an application with minor functional improvement. It is available at
937 no cost to a consumer, who already has the application resident on his/her phone.

938 b) A "patch" is implemented as an entire application.

939 c) A "patch" is associated with the "upgrade" price method.

940 d) Patches need to be explicitly added to the carrier ADS catalogs and explicitly
941 downloaded by the consumer.

942 e) A patch is a distinct Appl ID/version associated to the same part number as the
943 application to be "patched".

944 f) MobileShop will dynamically determine the application "patches" which
945 correspond to the applications residing on the phone.

946 g) When downloading the patch, the download will "overlay" the previous
947 application but the application user settings (i.e., highest scores, etc.) will be
948 retained. A patch will not affect the application usage settings associated with the
949 original download transaction.

950 h) Patches correspond to the "Upgrade" pricing method, with the same part number,
951 and no cost to the consumer.

4.1.14 Application Upgrades

a) An "upgrade" is an application associated with a new part number, with functional enhancements, and may optionally have an associated cost. It corresponds to the "Upgrade" pricing method.

b) MobileShop will dynamically determine the application "upgrades" which correspond to the applications residing on the phone.

c) Similar to patches, when an upgrade is downloaded, the previous application files are overlaid but the application user settings and application usage will be retained.

4.1.15 Catalog Language Support

This section identifies the language requirements as it relates to the catalog management functions.

4.1.15.1 Language of the Phone

a) The phone will interface to the ADS and pass the ADS criteria which will filter which applications are displayed to the consumer. The phone criteria which is passed to the ADS includes: platform, language (i.e., of the phone), and BREW API.

b) A phone may support multiple languages.

c) A phone will only operate in one language at a time.

d) If the phone supports multiple languages, the consumer may switch between languages thru a language select option.

e) Based on the language of the phone criteria,

- Catalog categories will display in the designated language
- Application preview text will display in the designated language
- Only applications in that designated language will be displayed in the catalog to the consumer.
- Applications in languages different than the specified language which are in the carrier ADS catalog, will not be displayed to the consumer.
- If the language is un-supported/un-recognized, the ADS will return the catalog in a "default" language.

4.1.15.2 Language of the Catalog

a) The catalog shall be able to support a category structure which can be displayed in different languages, but one language at a time.

b) For each category, the carrier shall be able to specify the category name in multiple languages.

4.1.15.3 Language Codes/Encodings

a) There may be more than one character set that needs to be supported per language on the phone.

b) A language code is the combination of language and character set.

c) Catalogs shall support multiple language codes.

d) For each category name, the carrier shall be able to specify the category name in multiple languages and language sets.

e) Platform ID will specify character set.

f) Applications will support specific languages and encodings. Refer to the Certification Extranet Specification for the list of languages/encoding values that an application may support.

4.1.15.4 Multi Language Applications in a Single Catalog

a) A carrier catalog may include single applications which can support multiple languages. Specifically, an application (i.e., single executable) may be developed to dynamically configure to run in different languages.

b) For Phase 1, each application (i.e., Appl ID) will be associated with one language which is configured in the application CAT file (Refer to the ADS specification). If a single application supports many languages it will be instantiated with as multiple Appl IDs with language specific metadata (i.e., Application name, description, etc.)

4.1.15.5 Hybrid Applications

a) A hybrid application is an application written in one language (i.e,. English) but available to a phone which is set to a different language (i.e., Japanese)

b) The carrier may accept hybrid applications in their catalogs.

c) In this case, the developer will be required to provide a minimum set of language specific attributes (i.e., application name, application description, preview text, etc.) in the "phone language" for the hybrid application.

d) The carrier has the option to include or exclude these "hybrid" applications in their catalog.

4.1.15.6 Application Display Name in the Catalog

Since each application is associated to one language, the application name will be consistent with the language of the application.

4.1.16 Adding A New Application to the Catalog

When adding a new application to the catalog, the following guidelines apply:

a) An application can appear across multiple categories within a single catalog.

b) An application that appears across multiple categories has the same application name and metadata as defined by the developer.

c) An application can only be listed within a single category once.

d) In Phase 1, the pricing options will display consistently for an application accessible from multiple categories within a single catalog version.

e) In the future, the pricing plans may differ for the same part number in different carrier catalogs (i.e., regional price plans for the same part number may vary)

4.1.16.1 Applications with Fixed Date Usage

a) One of the pricing basis allows for the definition of a fixed date for expiration of the application. The system shall provide proactive methods for monitoring these applications for the carrier. Refer to the section on UAM notifications.

b) In the future, If an application is configured for a fixed date and resides in an active or pending catalog, when the current date is beyond the fixed date, the application should not be available for a consumer to download the application for the associated price method(s). The application remains available for download if there are price bases other than fixed date defined.

4.1.17 Application Detail Information

UAM shall store application metadata by application, ISV and version. Refer to the Carrier Extranet Specification for further identification of application metadata upon application certification submission. This same information will be available to the carrier over the carrier extranet.

4.1.18 Platform Detail Information (Vicki)

UAM shall store platform metadata. Platform metadata shall include TBD:

4.1.19 Price Plans

a) The UAM will store the price plans associated to part numbers.

b) The carrier shall be able to view the pricing information by part number.

c) The carrier and developer may negotiate a change to the DAP.

d) All changes to the DAP must be submitted by the developer (i.e., via the developer extranet).

e) If an application is already contained in a carrier catalog, the developer must provide the carrier sufficient notice of any changes to the price plan including DAP. Refer to QIS Price Plan specification for further details on price plan, price templates and price locking..

4.1.19.1 Purchase Price Currency

The purchase price of the applications will be set in the currency of the catalog. For Phase 1, the catalog currencies to be supported include:

a) Korean Won

b) Japanese Yen

c) Canadian Dollar

d) Peso

e) U.S. Dollar

4.1.20 Preview Text

a) Preview text is the application description which shows up on the phone.

b) Developer enters the application name and the preview text on the developer extranet for each anguage encoding the applications supports. After certification, the ISV can edit the language specific name and text on the Developer Extranet.

c) When an application is added to a catalog the preview text is copied to the catalog specific list_name and description. The carrier can then edit the name and description further for that specific catalog. New or cloned catalogs will pick up the latest developer version of the text and the carrier can then add there own edits on the Carrier Extranet

4.1.21 Catalog Management Roles & Privileges

There are three administration privileges associated with catalog management as depicted in Figure 6.

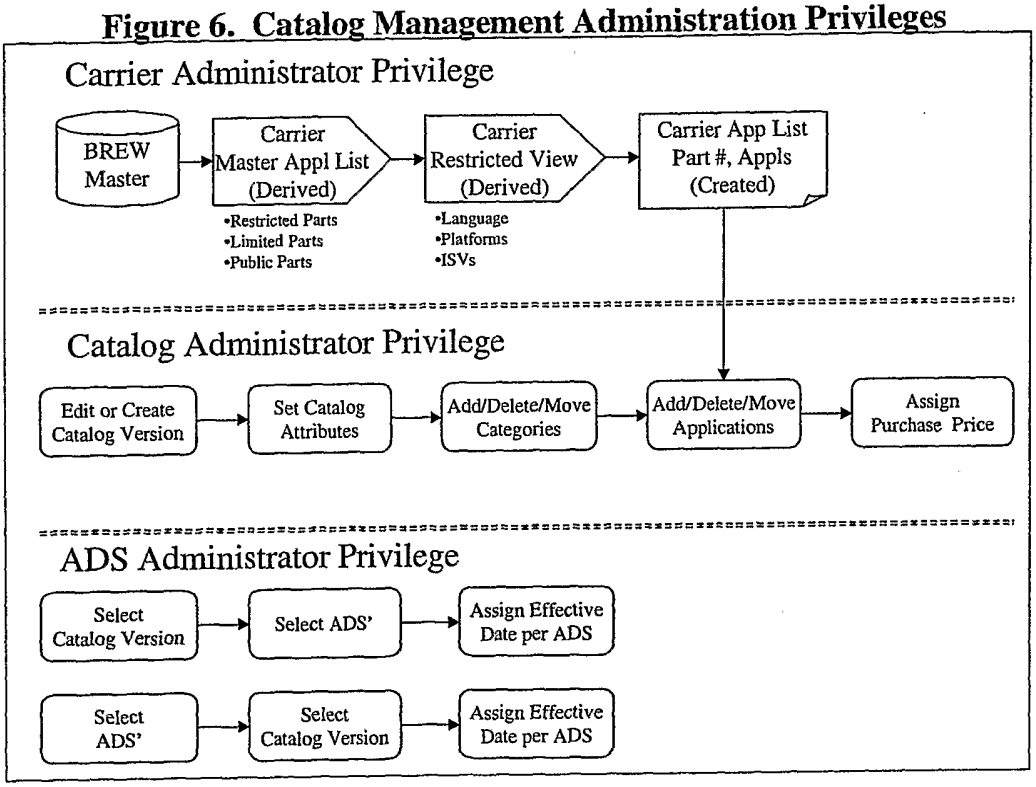
a) The carrier administrator is a super-user for catalog administration. This includes:

- The carrier administrator manages the Carrier Parts List.
- The carrier administrator review price plans and may collaborate with the ISV on a carrier specific price plan. Refer to the QIS Price Plan specification for more details in price plan collaboration.
- The carrier admininstrator may specify if a part number purchase price is editable by the catalog administrator.
- In addition, the carrier administrator has all privileges associated with catalog administration and ADS administration.

- Catalog Administrator manages catalog versions
  - The catalog administrator can create, clone and edit carrier catalogs
  - The catalog administrator and add/delete/order categories and applications within a catalog
  - The catalog administrator can set the catalog "READY" for pushing to an ADS.
- ADS Administrator manages the activation of catalogs on ADS farms.
  - The ADS administrator assigns catalog versions to ADS farms and the associated effective date for propagating the new catalog.

b) The primary carrier will identify named users to manage their carrier catalogs.

Figure 6. Catalog Management Administration Privileges



4.1.22 Audit Trails

a) All changes to the carrier catalogs shall be audited to a user level. This is important as catalog changes may affect application distribution and pricing options. From a support perspective, there will be a need to re-create history to reconcile support issues and support financial audits (i.e., revenue assurance).

4.2 Application Management

4.2.1 Sign-up as a BREW ISV

Prior to submitting applications for certification a developer company must complete a sign-up process for becoming a BREW Independent Software Vendor (ISV). UAM will store ISV metadata collected during the sign-up process for becoming a BREW ISV. Refer to the Developer Extranet section for further details on the web front-end related functions for ISV sign-up.

4.2.1.1 Authorized ISV Users

UAM will enable an ISV user to register a list of authorized developer contacts for the ISV. This list of developer contacts will subsequently be provided access to the developer extranet services.

4.2.1.2 Billing Attributes

UAM will store the ISV billing attributes required to process developer fees.

4.2.1.3 Definitive Agreement Confirmation

UAM will store the confirmation date/time stamp and the username in the ISV who confirmed the Definitive Agreement.

4.2.2 Application Submission Support for Certification

a) UAM will support the processing of the applications, documents, and metadata for submission to the BREW certification process.

b) Initially NSTL will be the first Application Certification Center (ACC) for BREW. Other ACCs may participate in the BREW Certification testing in the future.

c) These files will be delivered to QC via NSTL through a zip file.

d) The NSTL UAM interface will process the application and metadata into UAM,

e) In addition, the NSTL UAM interface will process applications documents into the UAM document repository.

f) The zip file delivered from NSTL will be stored as received in the UAM document repository, as an archive copy.

g) Once the submitted application is in UAM, it is stored as status pre-certified and restricted only to the carrier ACCHQ. ACCHQ carrier users can then add the application to their catalog and propagate the application to their ADS for certification OTA testing.

Refer to the Certification Extranet specification for further details on the ACC to QDC interface and related Certification extranet services.

4.2.3 ESN Management for Certification Testing

a) UAM will support the processing of Test enabled signatures for ACC ESNs .

b) The Test signatures will need to be applied to all applications in UAM which require certification testing.

c) There will be no system limitation enforced on the number of ESNs an ACC may need to generate test signatures for. This is unlike ISVs where there is a limit on the number of test signatures per ISV.

d) The ACC test signatures will have an expiration of 1 year or 365 days.

Refer to the Certification Extranet specification for further details on ESN management services.

4.2.4 Submission of Restricted Applications by Carriers

a) A restricted application is synonymous with a pass-thru application.

b) UAM will support the processing of the applications, documents, and metadata for restricted applications coming from carriers. Refer to the section on Carrier Extranet.

c) The carrier is to provide QC a part number for the restricted applications.

d) These files will be delivered to QC via TBD method.

e) Once the application is in UAM, it is stored as certified_status pass-thru, it is marked "ready for distribution" by ACCHQ and restricted only to the originating carrier.

f) The carrier can then add the application to their catalog and propagate the application to their ADS for commercial distribution.

4.2.5 "Ready for Distribution"

The ACCHQ shall mark standard applications "ready for distribution' via BREW Admin.

Only applications marked "ready for distribution" are available to carrier catalog management services.

Before an application can be marked "ready for distribution" it must have the following:

a) Certified By = <Carrier Name> or NSTL (Carrier name indicates a pass-thru application)

b) Certified Status = PASS or PASS_THRU

c) Part Number Assigned

d) Price Plan Associated with the Part Number

e) Billing Attributes (may be part number specific by carrier or use default settings for part number)

f) Export compliance forms where either filled out or on file.

g) Verify that the .mif file was renamed with the module ID ... ie: 905.mif

h) Verify that there is a .sig file.

i) Define Language specific application names and preview text.

4.2.6 ISV Application Distribution Features

a) An ISV has the ability to specify what carrier(s) are authorized to distribute their application. This means the ISV may select to include or exclude specific carriers per application. However, even if the carrier has authority to distribute the application, there is no guarantee that the carrier will or has added the application to the carrier's active catalogs.

b) For Phase 1, if an ISV decides to revoke a carrier's ability to distribute an application, the carrier is responsible for pushing a new catalog out to the carrier ADS which excludes the ISV application.

c) In the future, if an ISV decides to revoke a carrier's ability to distribute an application, the following will occur:

- The carrier master applications list would no longer contain the application and therefore the carrier will not be able to add the application to any new catalogs.
- All active and pending catalogs for the carrier will be scrubbed for the part number and removed/disabled from the carrier catalog.

1196 • The carrier will be notified of all system generated updates to carrier catalogs.

1197

4.3 Application Recall List

Application recall is a multi-step process that includes both manual and system functions. Application recall is a carrier specific recall request to remove a specific application/version from the phone users based on a date range period. The ACCHQ approves all application recall requests from carriers and processes the request through BREW Admin screens.

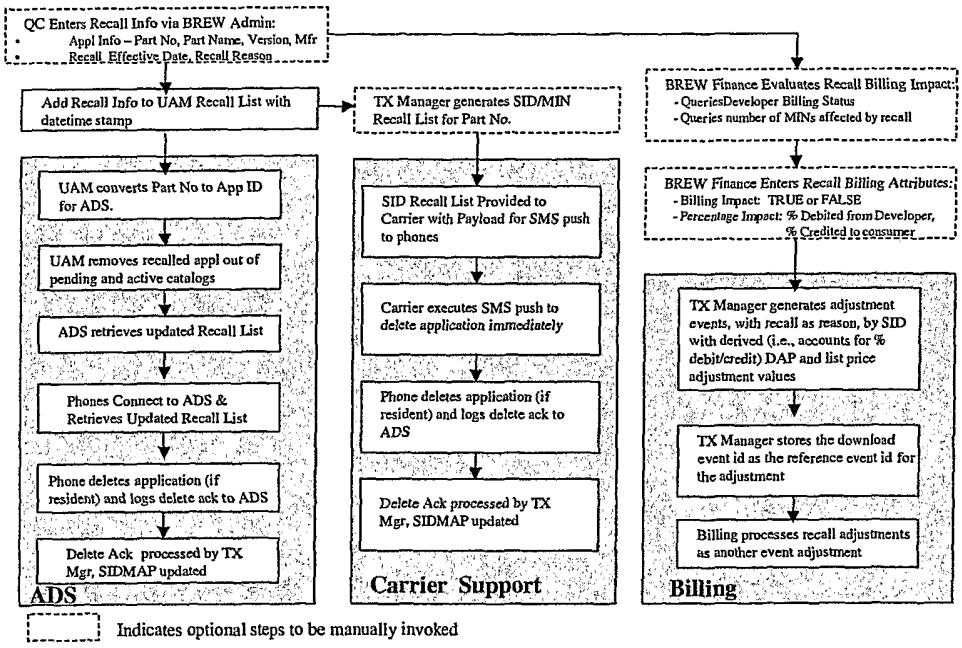
Application recall may be divided into the following functions:

- ADS Related Functions
  - This includes storing the application delete list in UAM and propagating the delete list to the ADS.
  - Once the application delete list is on the ADS the next time the phone connects to the ADS the applications on the phone will check against the delete list and applications will be removed automatically.
  - In addition, UAM will clean out any application contained in the carrier delete list from all carrier active and pending catalogs.
  - The catalog clone function will also compare against applications in the delete list so that these applications are eliminated from any cloning functions.
  - Note: Applications on the delete/recall list will only be removed the next time the consumer connects to the ADS. It is not guaranteed to be immediate.
- Carrier Support Functions
  - QC will provide the carrier with a min list generated from the UAM SIDMAP for application/versions to be recalled or deleted.
  - The MIN list will be used by the carrier to create an SMS payload message and "proactively" push a message out to the phone OTA to remove the malicious application.
  - This is an optional step and is up to carrier to decide whether this is required. It does enable an immediate removal of the application without requiring the consumer to connect to the ADS.
  - The next time the consumer connects to the ADS the delete acknowledgment is sent to the ADS and subsequently to TXN.
- Billing Functions
  - Depending on the nature of the recall, billing support functions will be provided in the recall process to add transaction adjustments into TXN of type "recall"
  - Billing implications for an application recall need to be reviewed thoroughly before executed as the implications may be large
  - If billing impacts are to be invoked, transaction adjustments will provide the mechanism to credit consumers and debit ISVs.

- The adjustment value to be derived for consumer credit and ISV debit can be configured by a percentage (i.e., 50%, 100%).

Figure 7 illustrates the Application Recall process described above.

Figure 7. Application Recall Processing Thread



4.4 Denied Party List (DPL) Verification

4.4.1 General DPL Requirements

a) To meet export compliance requirements all carriers, ISVs and OEMs that QC is performing business with must be checked regularly against a Denied Party List (DPL) currently maintained in AES.

b) If a company that QC is doing business with internationally, becomes a confirmed "hit" against the DPL, QC must suspend business with the company.

c) Export compliance requires e-mail on all possible "hits" against the DPL.

d) Export compliance will confirm the "hit" or identify it as a false hit.

4.4.2 Qadmin DPL Check

Since all of the BREW related organizations and accounts are managed through Qadmin and Qadmin has implemented a DPL check against AES, this meets the needs of the export compliance check.

Below is a brief summary of the DPL check implemented in Qadmin:

- Denied Parties List - new accounts will be inactive until verified against the "bad guy" Denied Parties List.
- The verification process will run every five minutes and will return a confirmation message.
- There will also be a process that runs once a day to compare the DPL against the user accounts.
- Any deltas will trigger an email to the Admin of the org that user is in so they can provide oversight and manually inactivate the user account.

4.4.3 Confirmed DPL Hit Implications

If an organization generates a confirmed DPL check, Qadmin will immediately disable the organizational accounts.

BREW needs to determine how to handle the in-process transactions (i.e, billing, cleaning catalogs, disabling other services) This is TBD.

4.5 Time Zones

4.5.1 Company Time Zones

a) In Phase 1, the middleware will only handle one time zone offset per company. This includes carrier

b) The time zone offset is used when displaying or prompting a user for a date/time value. It is used in the Carrier Extranet for effective date/time of catalogs.

4.5.2 Carrier Time Zones

a) In a future phase, the system will be enhanced to handle multiple time zone offsets for a single carrier.

4.6 Company and Contact Management

UAM is manage a company information and company contacts for carriers, ISVs, OEMs and other customers. This is in addition to account/organizational information managed in Qadmin/LDAP.

a) The company information will include:

- Address, City, State, Country
- Postal Code
- Weburl
- Phone, Fax
- Timezone Offset

b) The contact information will include:

- Last Name, First Name
- Title
- Company
- Contact Type
- Address, City, State, Country Code, Postal Code
- Phone, Fax, E-Mail, Mobilephone

4.7 Identifier (ID) Management

The UAM shall generate and manage the identifiers defined in Table 3.

Table 3. IDs Generated by UAM

Identifier	Requested By	UAM Request Method
Class ID	Developer	Developer Extranet
Applet ID	Developer	Developer Extranet
Log ID	Transaction Manager	Loadtxn interface from ADS
Module ID	Application Tool	BREW Admin
Package ID	Application Tool	BREW Admin
Platform ID	OEM	OEM Extranet
Carrier ID	Carrier	Carrier Set-up/Implementation, carriers are being implemented in blocks of 100.
Part Number	Product Support	Application Submittal to Certification Center
Company ID	Customer Support	New company in Clientelle. Company ID used to verify personnel who can request support from a particular company. Also to be used by Verizon to identify ISV seeking carrier guideline information.

4.6 UAM Roles and Security

Refer to the Web Services section on Roles and Privileges for further description functional requirements.

4.7 UAM Interfaces

UAM provides APIs to other BREW applications and services. Refer to the QDC UAM specification for further details on the interfaces identified below:

a) ADS interfaces for transaction log upload and application and catalog download to UAM (Refer to Section 6 discussion of the Carrier Extranet and the ADS Specification)

b) Web interfaces for developer, OEM, certification center, BREW admin, and carrier extranet services (i.e., UAM identifiers and application metadata)

c) NSTL interface for processing pre-certified application into UAM to include application metadata, documents and the application itself.

d) Clientele interface for replication of pre-developer and company information from UAM to Clientele.

4.8 UAM Notifications(Vicki)

UAM shall provide e-mail notifications for the following events:

a) ISV is notified when the application has been made available to the Certification Center

b) ISV is notified when the application is made "marked for distribution" by the ACCHQ.

c) ISV is notified when the application has been added to the BREW master, post-certification.

d) Carrier is notified when the ISV schedules the propagation of a modified DAP of an application currently included in an active or pending carrier catalog. Carrier is provided the revised DAP and the target effective date. The carrier will determine whether to continue distribution of the application or revise the application purchase price, if necessary.

e) Carrier should be notified if there is an application in an active or pending catalog with a price basis configured for a fixed date which is within 30 days of the current date.

5 Middleware Tools

5.1 Application Manager Tool

5.1.1 Process new application into UAM

The following are the steps for adding a new application into UAM using the Application Manager Tool:

a) Select "add new application" function

b) Enter all application metadata info, note the "Min API" field must contain version in format x.y.z --at least two dots.

c) Select checkbox for "New Module".

d) Select "browse" to go to your local directory where your application files are saved, and select the zip file that contains the application files. It is important to note that the zip file must not contain any sub directories (i.e., the .mif, .bar, .mod files must be in the top level of the zip).

e) "Save" to complete.

5.1.2 Process upgrade application into UAM

The following are the steps for adding an upgrade application into UAM using the Application Manager Tool:

a) Find the original application in the appmgr home and click on the package id link under column "package". Note the moduleId displayed in the page that comes up.

b) Go back to appmgr home and click on "Create new Application". Enter application metadata. Make sure the "new module" checkbox is NOT checked. and select the module ID you noted earlier from the select list.

c) Browse to your application zip file and select it.

d) "Save to complete".

5.1.3 Add an application with multiple packages into UAM

The following are the steps for adding an application with multiple packages into UAM using the Application Manager Tool:

a) Follow 5.1.1 to create an application with one of the packages. Then at the appmgr home, click on "edit" in the front of the application's row.

b) Select "New Module" checkbox.

c) Select "Browse" to select the zip file containing the second package.

d) Select "Save".

e) Repeat above steps until all packages are added.

5.1.4 Edit Application Metadata

The following are the steps for editing application metadata using the Application Manager Tool:

a) Select "edit" icon in the front of the application to edit.

b) Fill in different values as desired.

c) DO NOT modify the module section.

d) "Save".

5.1.5 Change Application Certification Status

The following are the steps for changing the certification status using the Application Manager Tool:

a) Edit application.

b) Enter the certification agency (Company name) in the "certified by" box by selecting the right value from the list.

c) Enter the certified date by clicking on the calendar icon and select the correct certification date.

d) "Save".

5.1.6 Application Signing

The following are the steps for signing an application using the Application Manager Tool:

a) Select the package id link under the "Package" column for the application you want to sign. You will see the page with "package files" displayed.

b) Select "check out" button. A "Download Package" button will be displayed.

c) Select "Download Package" button to download the application.

d) Create the appropriate signature for this application and name it "appname.sig".

e) Return to this "package files" page and see if there is already a file named "appname.sig", if so, click on "delete" button next to it to remove this file from the package.

f) Return to this page and click on "check in". An input box with a Browse button will be displayed. Use the browse button to select the "appname.sig" file you just created. Click "Upload" to add it to the package.

g) "Save".

5.1.7 Language Support

The following are the steps for managing language specific attributes using the Application Manager Tool:

a) Select the language link under the "LOCALE" column for the application.

b) Add language specific application information such as "native name" and "description".

5.2 Test Signature Tool

This section is TBD.

6 Web Services

This section defines the functional requirements for BREW related Web services. This includes:

- BREW Internet: BREW information and services available to the general public
- Carrier Extranet (CX): BREW services available to authenticated carrier users.
- Developer Extranet (DX): Extended developer BREW services available to users from authenticated developer companies.
- Certification Center Extranet (CERT): Certification centers services needed by external Certification Centers and QUALCOMM'S ACCHQ
- OEM Extranet (OX): BREW services provided to authorized device manufacturers.
- BREW Admin(BAD): An administrative tool used by QUALCOMM employees to manipulate UAM data, access queries and reports
- QAdmin: A tool developed by QC Corporate IT to manage organizations, groups and users, and the security permissions associated to those entities.
- QDOC: A tool developed by QC Corporate IT to manage the exchange of files between QUALCOMM and external entities.
- BREW Authorization Roles: Defines the mapping of BREW Roles to authorized functions.

6.1 BREW Internet

The BREW internet will provide public access to BREW information. The majority of information posted to the BREW internet is marketing information which will be maintained on a regular basis. The BREW internet shall be logically linked off of the QC external corporate site.

The types of internet services include:

- General marketing information targeted to developers, ISVs, carriers, and OEMs.
- Customer service functions targeted to developers, carriers, and OEMs.
- Links to other related web sites which will include: developer extranet, carrier extranet, OEM extranet and Certification Center extranet.
- BREW related events (i.e., Developer's conference)

Table 4 describes the functionality in each of these areas.

Table 4. BREW Internet Functions

Type	Functions	Sub-Functions
Marketing	About BREW	• About BREW – Press Releases – Fact Sheet – BREW in the Press – BREW White Paper • Events – Recent Events • Our Partners • Developer FAQ – Technical – Business – Export Compliance – Tax • Contact Us • Help • Site Map
Application Developers	Developer Overview	• Download SDK (see secure file exchange) • List/download SDK patches (see secure file exchange) • Technical Documentation On-Line (see secure file exchange) • Participate in Discussion Groups
	Become A BREW Developer	• Member Benefits – BREW Developer – Select BREW Developer – Elite BREW Developer • Developer Sign-up – SDK Download Registration – SDK Download
	Developer Support	• SDK Updates • Technical Documentation – BREW SDK User's Guide – BREW API Reference Guide – BREW Device Configuration Guide – BREW MIF Editor Guide – BREW Resource Editory Guide • Discussion Groups & Forums • Developer FAQ

		– Technical – Business – Export Compliance – Tax • Online Knowledge Base – How To – Advice – Info • Email Tech Support (brew-support@qualcomm.com) • Phone Types and Platforms (via BREW Developer Extranet) • Carrier Guidelines (via BREW Developer Extranet)
	Road to Market	• Developer Authentication (Verisign link) • ARM BREW Pack (via BREW Developer Extranet) • Developer Tools (via BREW Developer Extranet) – BREW Class ID Generator (needed for appl submit) – BREW AppSigner (needed for appl submit) – Grinder Tools – BREW TestSig Generator – BREW AppLoader • Developer Lab (schedule lab time through Developer Extranet) • Application Certification (via Developer Extranet) • Carrier Evaluation (Pricing Templates via Developer Extranet) • Carrier Acceptance (Developer Reports via Developer Extranet)
	BREW News	• Subscription Sign-up • Subscription Change (brew new e-mail address change) • BREW News Archives • Case Studies • Testimonials • Industry News • Events Calendar – Recent Events
On-Line Help	All Pages	• All pages should have on-line help

1435

6.2 Carrier Extranet

The carrier extranet will provide access to QDC services to authorized carriers. A user will be restricted to transact and view carrier specific information, where applicable. Some of the information on the carrier extranet is carrier specific and therefore restricted across carriers. Other information is general BREW information accessible to all carriers. A developer can create applications that will be restricted/limited for distribution by specific carriers. Other applications will be available for any carrier to distribute.

The types of extranet services include: authentication, catalog management, reporting, queries, customer service and secure file exchange. Table 5 describes the functionality in each of these areas.

Table 5. BREW Carrier Extranet Functions

Type	Functions	Sub-Functions
Authentication	Login	• Authenticate userid, password, token based
Carrier Admin	Application List	• Manage carrier application list • Filter applications based on application type, platform, language and developer. • Filter applications base on query type (available applications and price plans, carrier specific price plans, hidden carrier applications, currently carrier applications and price plans, expired carrier applications and price plans) • View application details and price plans • Expire an application and price plan from future catalogs • Add applications to carrier parts list. • Denote applications not yet included in catalog with special symbol
	Manage Application Price Plans	• List ALL applications available in Carrier Applications List • View Application Pricing Method and Basis Information • Within catalogs, the carrier can set the purchase price for each available pricing method. (i.e., displayed on the phone and can be different than Developer DAP). • The purchase price should be entered in the default currency of the catalog. For phase 1, a carrier has a single currency for list prices across all catalogs. • The carrier can select specific pricing methods and specific pricing basis from the developer's pricing plan to host in a catalog.
	Negotiate pricing w/ISV	• Review ISV price plans by part number • Agree to ISV pricing • Send email to ISV with requested carrier specific price plan modification
	ISV Information	• Query developer contact information by Part Number, Developer Name or Application Name
	Provisioned Price Plan	• Mark application for ISV to define provisioned pricing • Display provisioned pricing • Remove provisioned pricing from application
Catalog Admin	Create Catalogs	• Create, edit and delete new catalogs • Set catalog name, version, description, effective dates, languages, currency • Clone and edit all states of existing catalogs • Manage catalog states: pending, ready and active • Ready, Active and Deactivated catalogs may not be edited – they can only be cloned.
	Manage Catalog categories	• Create, edit and delete categories • Associate icons to categories • Modify sort order of categories • Enter additional language for categories
	Manage Applications within Catalog Categories	• Modify sort ordering of Applications within Categories (i.e., Move function) • Add, Delete Applications • Modify the ISV default preview text associated with the application

		• Within a single category, an application can have one price basis and pricing information for each pricing method. • Assign application level icons to be displayed on the phone. • The developer sets application names, but the carrier can append characters to them (future). • An application has the same name displayed across categories within a single catalog

1448

1448

Table 5. BREW Carrier Extranet Functions (continued)

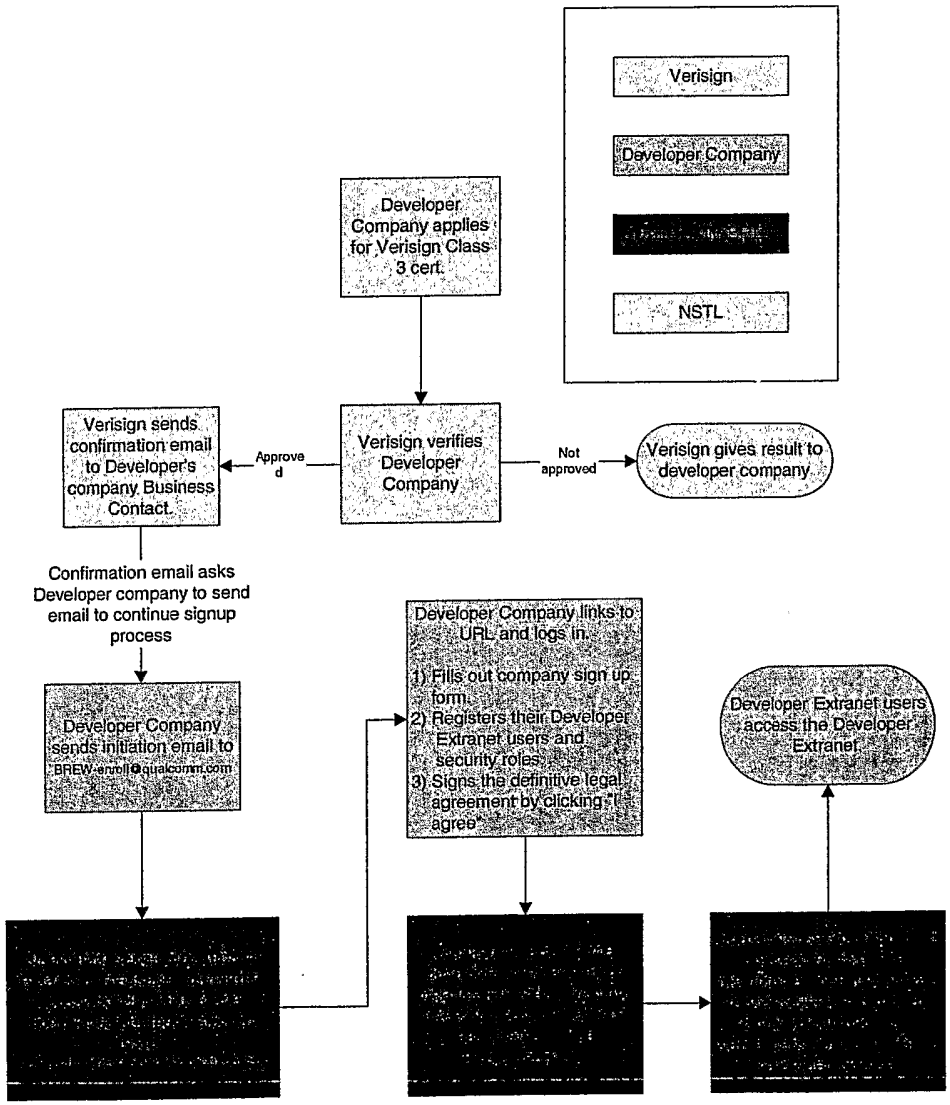
Type	Functions	Sub-Functions
ADS Admin	Application Availability on the ADS	• If a new catalog is pushed out to an ADS and a previously offered application is not offered any more, the application may still be used by subscribers who downloaded it previously. If a user had disabled the application and it is no longer available via the current catalog, it is still available on the ADS to be reloaded on that subscribers phone.
	Associate catalog to ADS'	• Specify carrier ADS to activate with this catalog. A catalog/version can be associated to multiple carrier ADS'.
	Catalog Version Activation Date	• Propagates all changes to the selected catalog version to the ADS at specified Activation date/time.
Carrier Reports – Catalog Admin	Catalog Report	• Query all states of catalogs (pending, ready , active and deactivated) • Select Catalog and Version and see contents.
	BREW Application Cross Reference Report	• Listing of all BREW Application and which are active on any Carrier's ADS. List by Carrier ADS ID.
	Activation Report	• Listing ordered by date descending, of all changes submitted for a Test catalog or Prod catalog for a specified ADS ID. Should list versions of catalog applied over a date range and the changes to the catalog applied at the activation date (i.e., application add/delete, pricing change, userid submitting the change, etc.).
Carrier Guidelines	Manage Carrier Guidelines	• A carrier maintains and publishes Carrier specific guidelines which are used to inform developers about carrier specific application attributes and pricing information.
Developer Info	Query for Developer Information	• Query developer contact information by Part Number, Developer Name or Application Name
Restricted Application	Submit	• Submit Application • Create Price Plans • Modify Price Plans
Billing Support	Transaction Adjustments	• Enter transaction adjustments for customer dissatisfaction, duplicate download, or accidental download
	MIN Updates	• Enter MIN deactivate and MIN transfer transactions
	Billing Extract Report by AR Invoice ID	• Generates a carrier billing extract file for transactions that correspond to an AR invoice ID. This is a batch report which may be delivered through QDOC.
	Carrier Invoice Report for ISV Fee	• Generates developer summary information for AR Invoice ID relating to ISV fees
	Carrier Invoice Report for QC Fee	• Generates summary information for AR Invoice ID relating to QC fees
	Carrier Usage Summary Report	• The Carrier Usage Summary Report will enable the carrier to retrieve the number of SIDs that have performed at least one download and/or delete event within a specified period. This is a batch report which may be delivered through QDOC.
	Carrier SID Summary Report	• The Carrier SID Usage Report will enable the carrier to retrieve usage metrics by SID within a specified period. This is a batch report which may be delivered through QDOC.
On-Line Help	All Pages	• All pages should have on-line help

1449

6.3 Developer Extranet

The developer extranet will provide access to Developer Alliance program information and QDC services to authorized ISVs. The types of developer extranet services include: authentication, alliance program information, news, development services, certification information, customer service, and secure file exchange.

The following diagram shows the ISV authentication and sign up process:



1462 Table 6 describes the functionality in each of these areas.
1463
1464

Table 6. BREW Developer Extranet Functions

Type	Functions	Sub-Functions
Authentication	Login	• Authenticate developer id, password • Organizations need to be created in Qadmin for each developer company • User names and passwords are managed in Qadmin for developer extranet access • Developer security roles are managed in Qadmin for developer extranet access
Updated Developer Benefits	Updated Developer Benefits	• To upgrade e-mail contact-brew@qualcomm.com
Technical Support	Developer Tools	• BREW Class ID Generator (web tool to generate a unique class ID) • Request for Applet IDs and Class IDs (unprotected URL on the extranet) – Developers may request a single ID or group of IDs. – The first 200 are reserved for use by QC. – The ClassID is a 32 bit number – In the interim, requests for ClassIDs are brew-support@qualcomm.com • BREW AppSigner (downloadable tool) • The Grinder Tool (coming soon) • BREW TestSig Generator (web based tool to generate a digital signature). ISV's limited to the number of ESNs to generate test signatures for. Expiration date of test signatures will be 90 days. • BREW AppLoader (downloadable tool to enable developer to transfer an application to the handset from a PC)
	E-mail Technical Support	• Form to send e-mail to brew-support@qualcomm.com for submitting technical support requests.
	Developer Lab	• Form to send e-mail to brew-support@qualcomm.com for requesting lab time.
	Phone Details	• List Phone Types • For each type enable download details and download device emulator device files
	ARM BREW Pack	• Purchase and download the ARM BREW Developers Pack
	Developer Support	• TBD
Business Development	E-mail Business Support	• Form to send e-mail to brew-support@qualcomm.com for submitting payment status requests.
	Carrier Guidelines	• TBD
Marketing	BREW Developer Directory	• Web form to add ISV to BREW Developer Directory
	Marketing Support	• BREW Developer Logo Use (BREW Developer logo use guide) • PR Assistance (PR Template, Spokesperson Talking Points – coming soon) • Web form for Inquiries and requests regarding PR assistance or use of the BREW logo on the ISV's website
Business Operations	Auto-reply e-mail	• When ISV sends e-mail to brew-enroll@qualcomm.com, and auto-reply e-mail is sent to initiate the BREW ISV sign-up.
Business Operations	ISV Company Sign Up	• ISV company signs up to be a BREW authenticated developer company . The ISV needs to complete the following. – Complete Company Information – Submitter Information – Contact Information (Business Contact, Technical Contact) – Extranet Administrator Role – Developer Role – Marketing Role – Financial Role
	Submit Software Application	• Link to NSTL site, coming on Aug 15 • Legal Agreements, Pre-Certification Test Plans, Checklists
	Application Management	• Application Manager Display including application version detail screen • Export Compliance/Encryption • Default Price Plan Updates – Maintain default application price plan which may be applied across multiple

		carriers. – All edits should be associated with an assigned effective date. – If DAP changes are made and the part number with default pricing is contained in an active catalog(s), the carrier(s) must be notified. Carriers must be notified at least 60 days before the change takes affect. – Notify all carriers minus those with carrier specific price plans and subject to restrictions, when a new price plan is superceding a default price plan. – Refer to the QIS Price Plan Specification for further details in Price Plan requirements • Carrier Specific Price Plan Updates – Maintain carrier specific application price plan. – All edits should be associated with an assigned effective date. – If DAP changes are made and the part number with carrier specific pricing is contained in an active catalog(s), the carrier must be notified. Carriers must be notified at least 60 days before the change takes affect. – Refer to the QIS Price Plan Specification for further details in Price Plan requirements • Carrier Price Plan Collaboration – Enable the ability for ISVs and Carriers to negotiated price plan configuration thru on-line collaboration. • Carrier Limited Application Distribution – Maintain the subset of carriers to distribute the application to if not to all carriers – If the developer revokes a carrier's ability to distribute his/her application then the pending and active carrier catalogs will need to be modified to remove the developer part number. • Adding Provisioned Price Plan – Only can be created by ISV after Carrier activates provisioned plan for the part number – Included in standard price template functionality, just hidden • Query Active Catalogs – Query catalogs where developer part number is included. – Review price plan for the part number in a selected carrier catalog. • Carrier Contact Information – Retrieve carrier contact information to discuss application price plan settings. The carrier must present any contact information beyond that provided in baseline Developer Extranet specification through the Carrier Guidelines. • Maintain Application Metadata – The developer needs to add the default name and description for every language his application supports when he fills out the price plans after certification. There needs to be page(s) to allow the ISV to add maintain the metadata like language and description after the app has been certified.
	Reports	• Manufacturer's Receivables Report – The purpose of the Manufacturer Receivables report is to provide the developer additional payment information relating to their developer fees processed by QC. The report is retrieve based on invoice ID (i.e., AP invoice ID). • Carrier Billing Report – The purpose of the Carrier Billing report provides the developer a view into the carrier billing corresponding to the developer's application usage. • Application Usage Report – The purpose of the Application Usage report is to provide the Manufacturer a view into application transactions by price method. • Carrier Term Sheet – TBD <Open Issue: isn't this the Guideline?>
On-Line Help	All Pages	• All pages should have on-line help

1465

6.4 Certification Center Extranet

The certification center extranet will be used to exchange data between the QDC and NSTL (National Software Testing Laboratory). The main functions will be to administer catalogs of test applications and push those catalogs to the NSTL ADS. Application test results will be maintain by NSTL and updated in the UAM. Refer to the QIS Certification Extranet Specification for functional requirements specifics.

6.5 OEM Extranet

The OEM extranet will provide access to QDC services to authorized OEM users. Table 7 describes the functionality to be provided over an extranet.

Table 7. BREW OEM Extranet Functions

Type	Functions	Sub-Functions
Authentication	Login	• Authenticate developer id, password
OEM Information		• Web form for OEM to manage General Information and Contact Information
Application Management	Provisioned Applications	• Displays list of applications/versions available to OEM • Enables OEMto download application(s) for pre-installation on the phones
	Restricted Applications	• Submit OEM Application • Set Price Plan
Platform ID Administration		• Create new BREW Platform IDs (Open Issue: should this be a request to create?) • Modify existing Platform IDs (i.e., modify platform information and platform properties)
OEM Support		• E-mail to brew-support@qualcomm.com • Display OEM customer support phone numbers • Links to (FAQs, BREW API documentation, Porting Guide User Guide, SDK Documentation)
OEM Tools		• Porting Kit download • BREW API download • OEM specific SDK Patches • OEM specific BREW Patches • Sample test plans • Diagnostic (app to load on phone to exercise the BREW API) • Test sig file generator (via Developer Extranet) • Code signer (via Developer Extranet) • Grinder (via Developer Extranet)
On-Line Help	All Pages	• All pages should have on-line help

6.6 BREW Admin Intranet

The BREW Admin Intranet will provide access to QDC services to authorized QC users.

This section is TBD.

6.7 Account Management using Qadmin, Siteminder

Qadmin is an account management tool created and maintained by Corporate IT and used in the QDC middleware. QAdmin is simply a GUI interface for managing data in an

LDAP repository. LDAP is an acronym for Lightweight Directory Access Protocol, which is a set of protocols for accessing information directories.

For QDC extranet services, LDAP is used to manage company, group (i.e., role), user and administrator information. Its primary use in the extranet is for controlling access to extranets (i.e. websites) and extranet applications.

In conjunction with LDAP, Siteminder is used to enforce authentication for web URLs. Siteminder can be configured for username/password authentication or secure ID authentication per URL. The authorization component which is based on the group or role access is enforced in the QDC web applications.

Qadmin provides a user-friendly front-end to LDAP for managing the company, group, user and administrator information.

When a user logs into the QDC extranet services the following authentication checks are executed:

a) User is authenticated in via Siteminder/LDAP. Depending upon the QDC website URL, secureid authentication may be required.

b) If the user is not in LDAP, then the user gets an authentication failure

c) If the user is in LDAP and the user is not in the UAM PC_USERS table, the user record is created and inserted into the UAM PC_USERS table.

d) The user roles are queried out of Qadmin/LDAP and used by QDC web services to govern authorization to service thereafter.

6.7.1 Siteminder Set-up

a) QDC team moves web pages to BREWX

b) QDC team provides web URL(s) to set-up with username/password authentication

c) QDC team sends e-mail to web-server-admin to configure Siteminder according to (b) above.

6.7.2 Qadmin Account Management Procedures

a) E-mail qadmin-support to set-up new organizations designating Sue Wake and Rachel Murphy as ITA Admins

b) Corporate IT creates the new organization in LDAP

c) ITA admins set-up pre-defined list of BREW people as Functional Admins for each BREW related organization

d) E-mail qadmin-support for internal QC user account set-up – BREW Product Support

e) Create external user accounts – BREW Product Support

f) Create groups within organizations – Functional Admins

g) Manage user membership in groups – BREW Product Support

6.7.3 Change/Reset Password

Qadmin provides services for users to change their passwords and administrators to reset user passwords.

6.8 QDOC

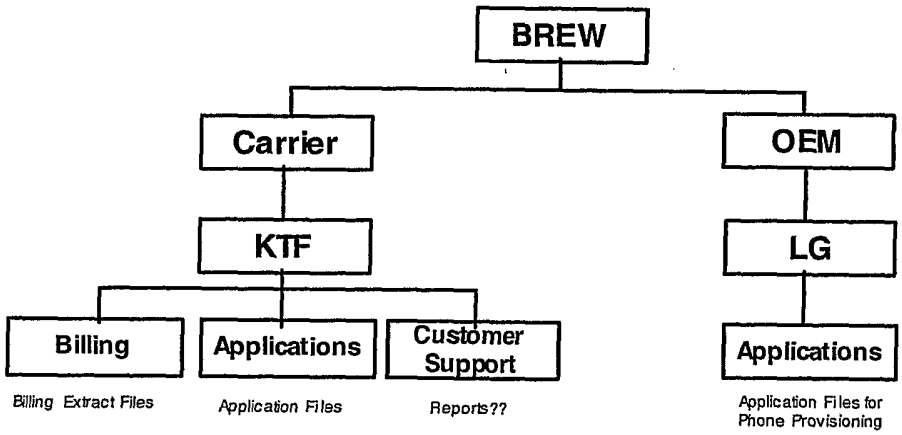
QDOC is an file exchange tool for the extranet, created and maintained by Corporate IT and used in the QDC middleware. It is meant for providing a secure environment for temporarily placing files for external entities to retrieve data from QC, and for external entities to temporarily post files for internal QC retrieval. QDOC is not a long term file storage repository. Access to workspaces and folders is based on groups.

BREW currently has a workspace defined in QDOC. Within the workspace, BREW can configure the structure to exchange documents with Carriers, OEMs and other external entities. External users will only be able to access BREW workstpace folder structures based on their group/role access. Each folder may be configured with the list of groups that can read or read/write to the folder.

QDOC is integrated with Qadmin. As such the user group membership is used to govern what permissions the user has in specific QDOC folders.

An example of the BREW workspace configuration is illustrated in Figure 9.

Figure 9. Example BREW Workspace Configuration in QDOC



6.8.1 QDOC Set-up

a) Define BREW workspace/folder/group set-up – QDC Team

b) E-mail qdoc-support to create BREW workspace reference group to BREW admin Workspace – QDC Team

c) Create folder structure within BREW workspace – QDC Team

d) Configure group access to folder structure – QDC Team

e) Configure e-mail notifications, as required – QDC Team

6.8.2 Change/Reset Password

Since the user account and password is integrated with Qadmin/QDOC to change password and reset password functions are implemented through QAdmin.

6.9 QDC Roles and Privileges

The following table identifies the QDC roles and associated privileges associated with each role. The roles and role membership are defined and maintained in Qadmin/LDAP. The applications govern the authorization aspects of what roles provide what privileges and the application level.

6.9.1 Carrier Extranet Roles and Privileges

Web Site	Category	Function	CX_ADM	CX_CAT	CX_ADS	CX_BIL	CX_ADJ	CX_RPT	CX_ISV	CX_APP
CX	Home	Home Page								
CX	App List	Add Applications/Price Plans	x							x
CX		Expire Applications/Price Plans	x							x
CX		Price Plan Collaboration	x							x
CX	Add New Catalog	Create a new catalog, set list price	x	x						
CX	Catalogs	Open/Edit/Clone Catalog	x	x						
CX	ADS Admin	Assign Catalog to ADS	x		x					
CX	Parts List	Display Carrier Parts List	x	?						
CX	Reports	Transaction Adjustments				x	ro			
CX		MIN/SID Updates				x				
CX		Billing Reports				x		x		
CX	Help	Customer Service Information	x	x	x					
CX	Developer Tools	Pass Thru Apps functionality							x	

6.9.2 BREW Admin Roles and Privileges

Web Site	Category	Function	BAD_QC	BAD_BIL	BAD_ADJ	BAD_RPT	BAD_ISV	BAD_APP	BAD_CRM

BAD	Home	Home Page	x	x			?	x	
BAD	Applications	Application Manager Tool	x			x	x		
BAD	Billing	Carrier/ISV/Part Number Billing Attributes (CARRER_MFG_FIN Table)	x	x					
BAD	Carriers	Update/Add Carrier (PRIMARY_CARRIER Table)	x	x					
BAD	Companies	Update/Add Companies (COMPANY Table)	x						
BAD	Download Servers	Update/Add ADS (ADS Table)	x						
BAD	Manufacturer	Update/Add ISVs (MANUFACTURER Table)	x	x					
BAD	Modules	Update/Add Modules (MODULE Table)	x						
BAD	Parts	Update/Add Parts (QC_PART Table, RESTRICTED_PARTS Table, MFG_PRICE_PLAN Table, MFG_PRICE_METHOD Table, MFG_PRICE_OPTION Table, SUBSCRIPTION Table)	x						
BAD	Platforms	Update/Add Parts (PLATFORM Table, PLATFORM_APPS Table)	x						
BAD	Reports	SDK Download Report	x					x	
BAD		Carrier Parts List Report	x					x	
BAD		Transaction Adjustments	x	x	ro				
BAD		Developer Adjustments	x	x	ro				
BAD		MIN/SID Updates	x	x					
BAD		Billing Reports	x			x			
BAD	Subscriptions	Update/Add Parts (SUBSCRIPTION Table)	x						

BAD	Users	Update/Add Users (PC_USERS Table)	x
BAD	Help Admin	Update/Add On-Line Help Text (CG_HELP)	x
BAD	Help	Customer Service Information	x

6.9.3 Developer Extranet Roles and Privileges

Web Site	Category	Function	DX_BIL	DX_RPT	DX_ISV	DX_VER	DX_APP	DX_CRM	DX_DEV
DX	ISV Sign-Up	Company Info				x			
DX		Company User IDs and Roles				x			
DX		Definitive Agreement				x			
DX	App Submittal	NSTL Link			x		x		x
DX	ISV Info	ISV Metadata Updates			x				
DX	ESN Mgmt	ESN/Test Sigs			x				x
DX	Price Plan	Create/Edit/Clone Price Plan			x		x		
DX		Price Plan Collaboration			x		x		
DX	Reports	Billing Related Reports	x	x	x			x	
DX	Developer Tools	Signing and testing tools			x				x

6.9.4 Certification Extranet Roles and Privileges

Web Site	Category	Function	CERT_ACCHQ	CERT_ACC
CERT	ACCHQ		x	
CERT	(ACC) NSTL			x

6.9.5 OEM Extranet Roles and Privileges

Web Site	Category	Function	OX_OEM
OEM Extranet	All	All	x

6.9.6 QDOC Extranet Roles and Privileges

Web Site	Category	Function	QC	BIL	APP	OEM	CRM	ENG	TRAIN
QDOC	Carrier Organization	Roles used under Carrier Folder Trees in QDOC		CX_BIL	CX_APP		CX_CRM		
	OEM Organization	Roles used under OEM Folder Trees in QDOC			OX_APP	OX_OEM			
	QC Organization	Roles used under Carrier Folder Trees in QDOC	QC	BIL			CRM		
	QC Organization	Roles used under OEM Folder Trees in QDOC	QC		APP			ENG	
	QC Organization	Roles under TBD Folder in QDOC							TRAIN

Appendix A Glossary

This glossary defines terms, acronyms, and abbreviations used in the document.

ACC	**Application Certification Center**
ACCHQ	**Application Certification Center Headquarters**
ADS	**Application Download Server**
BREW	**Binary Runtime Environment for Wireless**
CRM	**Customer Relationship Management**
DAP	**Developer Application Price**
EFT/ACH	**Electronic File Transfer/ Automated Clearing House**
ESN	**Electronic Serial Number**
GMT	**Greenwich Mean Time**
MIN	**Mobile ID Number**
OEM	**Original Equipment Manufacturer**
OTA	**Over-the-Air**
QC	**QUALCOMM**
QDC	**QIS Distribution Center**
QIS	**QUALCOMM Internet Services**
SID	**Subscriber ID**
SMS	**Subscriber Message Service**
SNF	**Store –N- Forward**
TXN	**Transaction**
UAM	**Unified Application Management**
VPN	**Virtual Private Network**
XML	**Extensible Markup Language**

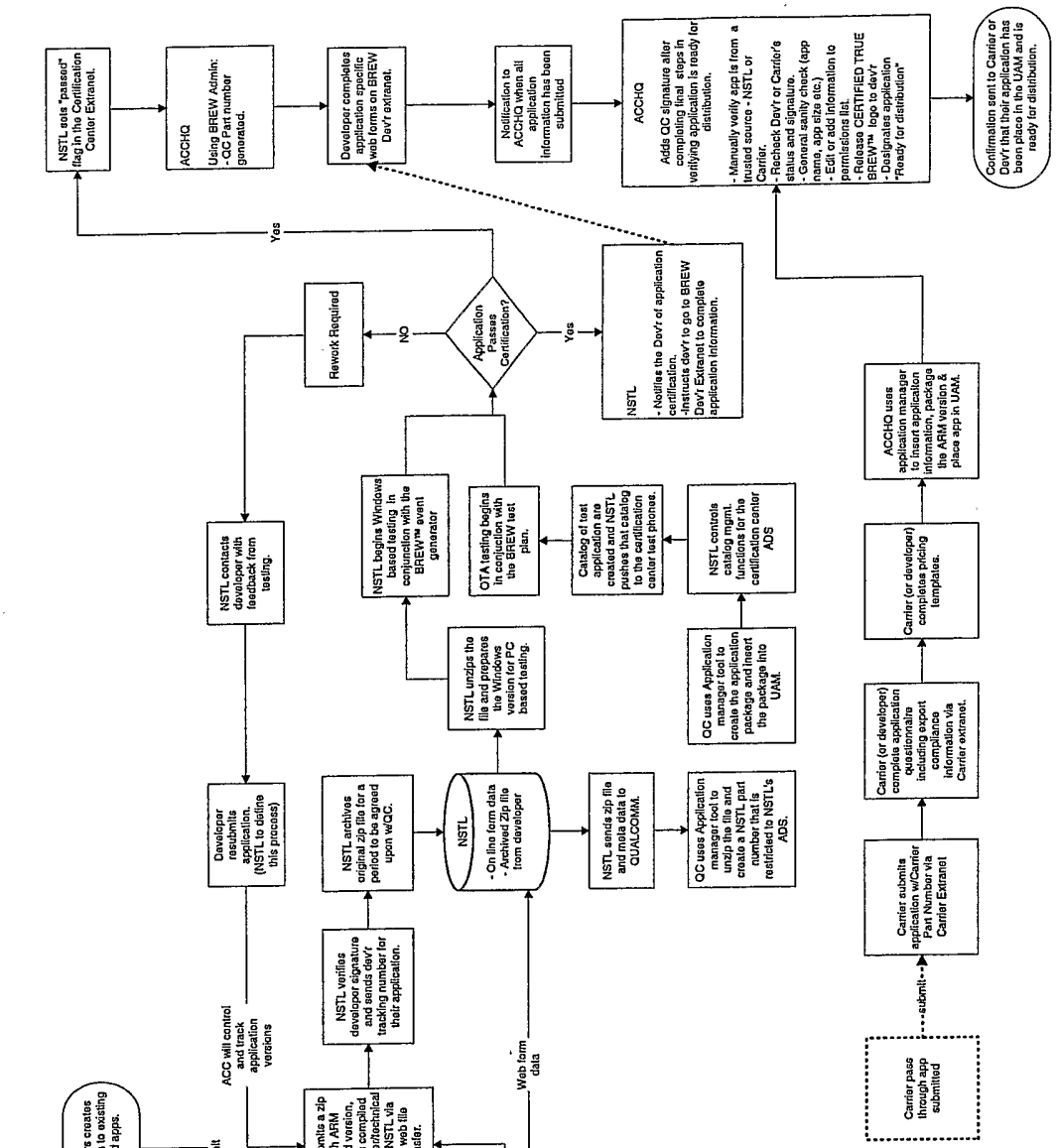
Appendix B – BREW Level 1 Process Definitions

This appendix contains the BREW Level 1 process definitions. The list of process definitions are identified in the following table.

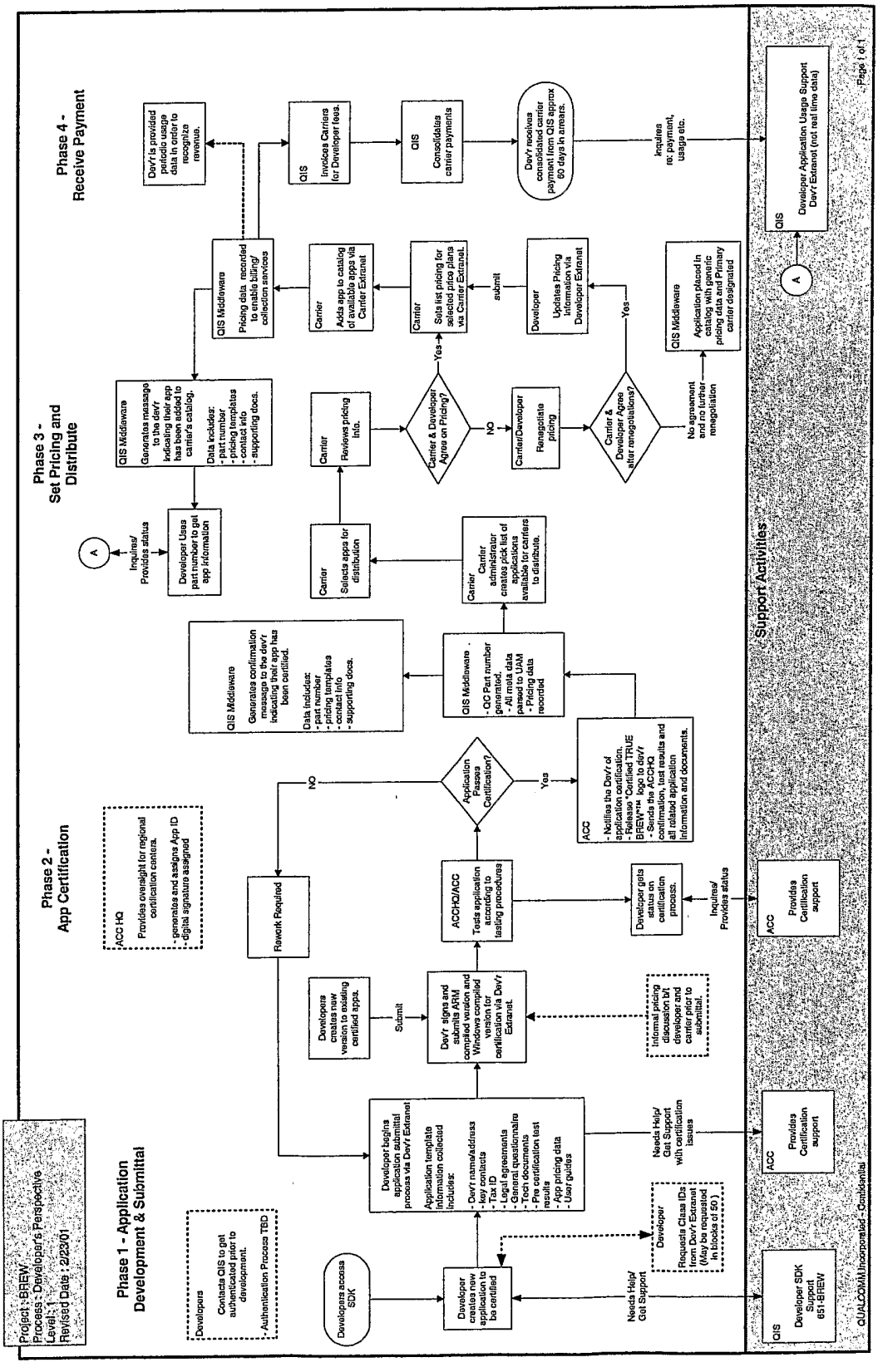
Level 1 Process Definition Diagrams

Level 1 Perspective:	Process Phases:	Document Location
Application Certification	Application Submittal Application Management & Testing Notifications Revoke an Application	Appendix B, Section 1.0
Developer	App Development & Submittal Application Certification Set Pricing and Distribute Receive Payment	Appendix B, Section 2.0
Carrier	Negotiate and Select Apps App Availability / User Trials Production Availability Accounting and Collections	Appendix B, Section 3.0
Consumer	User Invoked Activity Download (transact) apps Use & Pay for apps	Appendix B, Section 4.0
Distribution Center	Receive certified apps Build / Maintain Carrier Extranet Manage Apps via UAM Support OEMs Transact Financial Data	Document in Development

1583 **1.0 Application Certification Process**

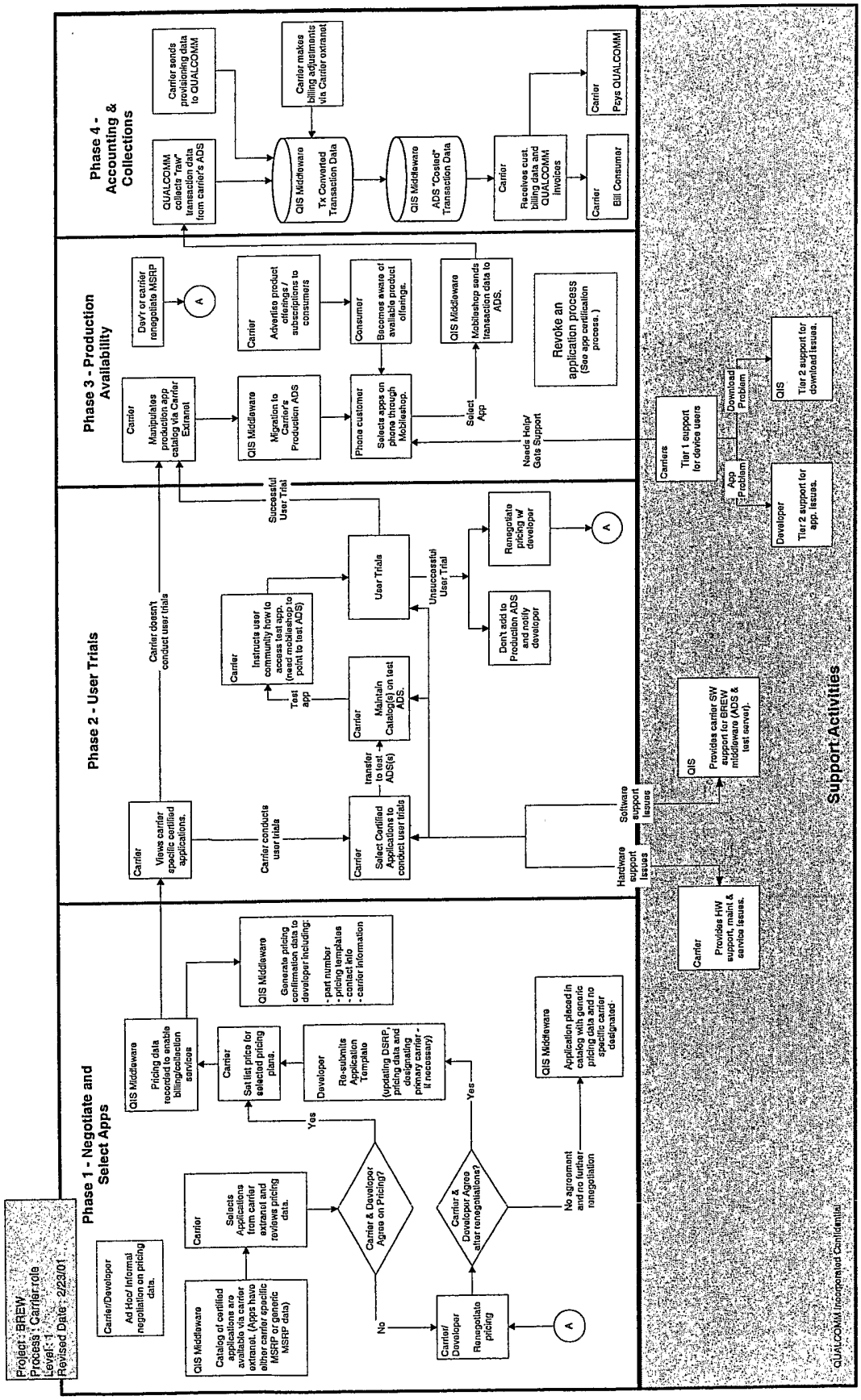


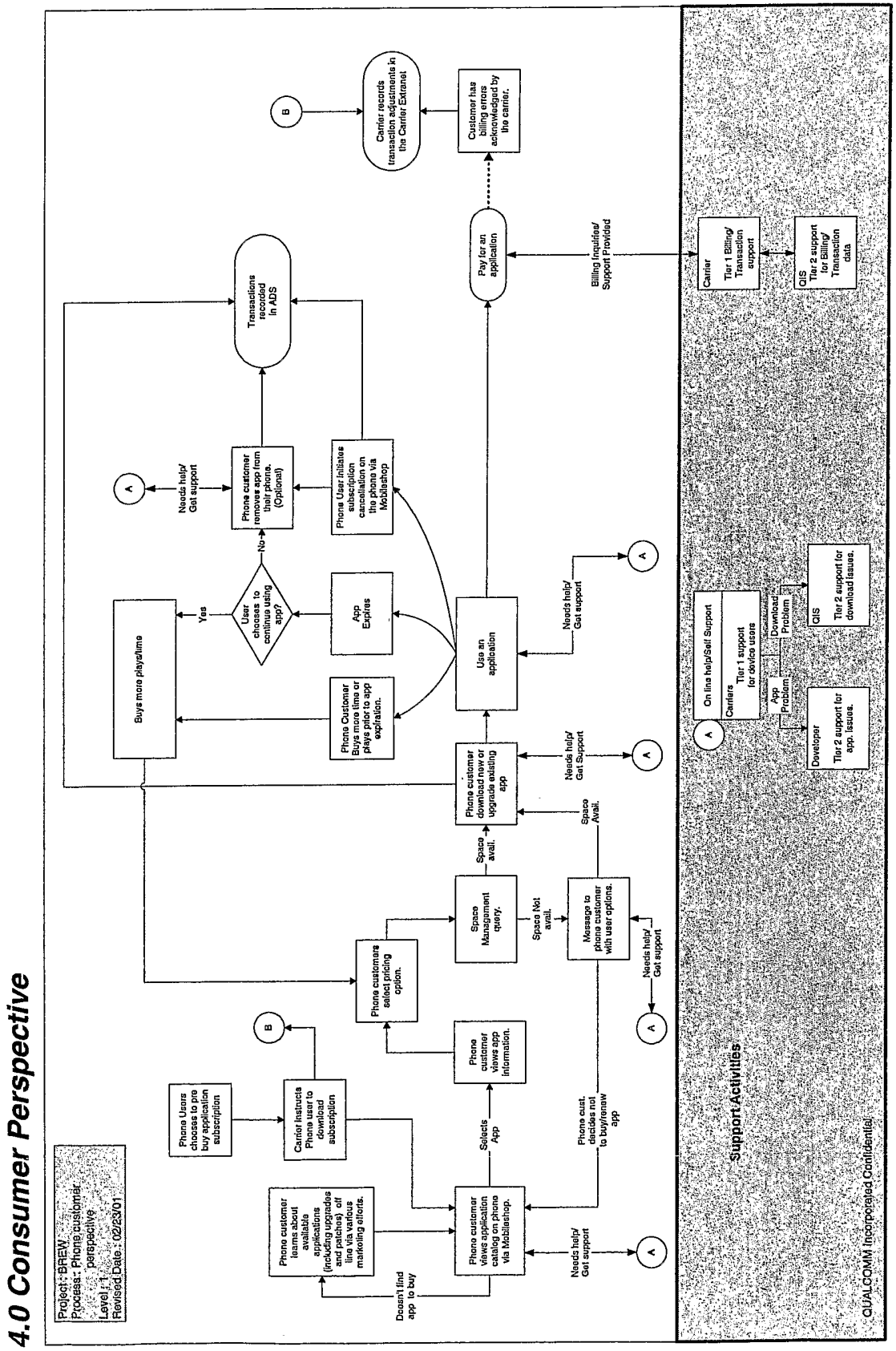
1584 **2.0 Developer Perspective**



1585 **3.0 Carrier Perspective**

1586




4.0 Consumer Perspective
Project: BREW
Process: Phone customer perspective
Level: 1
Revised Date: 02/23/01
Phone Users chooses to pre buy application subscription
Carrier instructs Phone user to download subscription
Phone customer learns about available applications (including upgrades and patches) off line via various marketing efforts.
Phone customer views application catalog on phone via Mobilashop.
Doesn't find app to buy
Selects App
Phone cust. decides not to buy/view app
Phone customer views app information.
Phone customers select pricing option.
Space Management query.
Space Not avail.
Message to phone customer with user options.
Space Avail.
Space avail.
Phone customer download new or upgrade existing app
Needs help/ Get support
Use an application
App Expires
User chooses to continue using app?
No
Yes
Buys more plays/time
Phone Customer Buys more time or plays prior to app expiration.
Phone customer removes app from their phone. (Optional)
Phone User initiates subscription cancellation on the phone via Mobilashop
Transactions recorded in ADS
Pay for an application
Billing Inquiries/ Support Provided
Carrier records transaction adjustments in the Carrier Extranet
Customer has billing errors acknowledged by the carrier.
Support Activities
On line help/Self Support
Carriers
Tier 1 support for device users
Developer
Tier 2 support for app. issues.
Carrier
Tier 1 Billing/ Transaction support
OIS
Tier 2 support for Billing/ Transaction data
QUALCOMM Incorporated Confidential

1586 1587

5.0 QDC Perspective

This process definition is TBD.

Appendix C – Encryption and Export Compliance

2.3.1 Encryption and Export Compliance

The developer is required to specify the level of encryption contained within the application. This section identifies the details of how the developer must identify the application encryption.

Developer will chose from a drop down list of four choices;

1. No Encryption
2. Encryption for Authentication/Password Protection
3. Encryption for mCommerce/eCommerce/Banking
4. Other Encryption

There will be a help button that will take them to the attachment "BREW encryption guidelines", refer to Appendix A

2.2.1.1 No Encryption

If developer chooses #1, a report is generated behind the scenes by QIS for export compliance.

2.2.1.2 Encryption for Authentication/Password Protection or Encryption for mCommerce/eCommerce/Banking

If the developer chooses #2 or #3, the developer is taken to a screen where they must fill out the following information. A report with this data has to be available by export compliance.

❑ **Type 1 – Encryption for Authentication/Password Protection OR mCommerce/eCommerce/Banking**
Software with limited cryptographic functions for access control, password protection or authentication. (There is no encryption of files or text except as directly related to the protection of passwords, Personal Identification Numbers (PINs) or similar data to prevent unauthorized access.)

The software ***is not*** user-accessible ***and is*** limited to or specially designed for **any** the following functions (select Yes or No for each item below if your item falls into this category):

Yes	No	
	□	Execution of copy-protected software,

- Copy-protected read-only media,
- Information stored in encrypted form on media where the media is offered for sale in identical sets to the public,
- One-time encryption of copyright protected audio or video data,
- Authentication or digital signatures,
- Fixed data compression or coding techniques,
- Banking or money transactions, or
- Password protection, PINs or data protection for preventing unauthorized access.

2.2.1.3 Other Encryption

If the developer chooses #4, **they have two choices:** (Vicki...only see one choice here.....)

❑ **Type 2**
Other cryptographic functions (not described above) for transforming data in order to hide information, prevent its undetected modification or prevent its unauthorized use.

For Type 2 software, a formal U.S. government commodity classification will be required before the software can be exported from the U.S. and made commercially available to the carrier. If you company has not applied for the classification please check here: ☐

If you have the commodity classification information, complete the information in the box below:

U.S Government BXA Classification Leave blank if unavailable.	ECCN:	License Exception:	CCATS #:*	CCATS Date:		
Was the encryption product granted RETAIL status by the U.S. Department of Commerce? (Circle one)				Yes	No	NA
* If applicable, send a fax copy of the CCATS classification to (858) 651-1767 along with this form.						

If they don't have a G number to enter above, they must fillout the encryption questionnaire form (Appendix B) and submit source code to the government:

BREW Encryption Guidelines

QUALCOMM®

U.S. Export Compliance General Information

The U.S. Government's Bureau of Export Administration regulates exports of products and technology, including software offered for download outside the U.S. To determine the export requirements of your software application, the technical characteristics of the software must be reviewed against U.S. government guidelines.

These technical characteristics will fall into one of the three following categories:

❑ **Type 1**
Software with limited cryptographic functions for access control, password protection or authentication. (There is no encryption of files or text except as directly related to the protection of passwords, Personal Identification Numbers (PINs) or similar data to prevent unauthorized access.)

The software ***is not*** user-accessible ***and is*** limited to or specially designed for **any** the following functions
(select Yes or No for each item below if your item falls into Category 1):

Yes No

- ☐ Execution of copy-protected software,
- ☐ Copy-protected read-only media,
- ☐ Information stored in encrypted form on media where the media is offered for sale in identical sets to the public,
- ☐ One-time encryption of copyright protected audio or video data,
- ☐ Authentication or digital signatures,
- ☐ Fixed data compression or coding techniques,
- ☐ Banking or money transactions, or
- ☐ Password protection, PINs or data protection for preventing unauthorized access.

☐ **Type 2**

Software with other cryptographic functions (not limited to Type 1 software above) for transforming data in order to hide information, prevent its undetected modification or prevent its unauthorized use.

☐ **Type 3**

No cryptographic functions.

Note: For Type 2 software, a formal U.S. government commodity classification will be required before the software can be exported from the U.S. and made commercially available to the carrier.

For additional information on U.S. government export controls visit the http://www.bxa.doc.gov/ or see the export compliance section of our Frequently Asked Questions at http://www.qualcomm.com/brew/support/kb/50.html.

1698 ***Encryption Questionnaire***

1699

)MMODITY CLASSIFICATION – ENCRYPTION QUESTIONNAIRE
; U.S. Government's Bureau of Export Administration regulates exports of products and technology, including software offered for download ;ide the U.S. To determine the export requirements of your software application, the technical characteristics of the software must be reviewed ıinst U.S. government guidelines. The below information is required to determine which export requirements may apply to your BREW)lication.

1700

1. List the symmetric and asymmetric encryption algorithms and key lengths and also describe how the algorithms are used (e.g., 56-bit DES, 168-bit 3DES, 128-bit RC4, 448-bit Blowfish, etc). Specify which encryption modes are supported (e.g., cipher feedback mode or cipher block chaining mode.):

Symmetric Algorithms			
Name of Algorithm	**Maximum Key Length**	**Mode(s)**	**Use**

Asymmetric Algorithms		
Name of Algorithm	**Maximum Modulus Size**	Use

Hash Algorithms (list names)	**Use**

2. State the key management algorithms and any key management protocols not listed above, including modulus sizes, which are supported (e.g., 512-bit RSA, 1024-bit Diffie-Hellman:

   Explain how the application uses these algorithms and protocols.

3. If using a proprietary algorithm not widely available, include a textual description of the algorithm. Attach the source code for review by the U.S. Government.

4. Describe any pre-processing methods (e.g., data compression [LZS, Deflate, etc.] or data interleaving) that are applied to the plaintext data prior to encryption.

5. Describe any post-processing methods (e.g., packetization, encapsulation) that are applied to the cipher text data after encryption:

3. List the communication protocols (e.g., X.25, Telnet or TCP) and encryption protocols (e.g., SSL, IPSEC, or PKCS standards) that are supported:

Describe these protocols and explain how the application uses them:
Attach the source code for review by the U.S. Government if the protocols are not widely available.

7. Does this software contain a general application programming interface (e.g., one that accepts either a cryptographic or non-cryptographic interface but does not itself maintain any cryptographic functionality)?
Yes ☐ No ☐ If Yes, describe and identify which are for internal (private) use and which are external (public) use:

3. Identify the third party libraries or other sources for the encryption functionality below:

Library	Manufacturer	Static or Dynamic

). For commodities or software using Java byte code, describe the techniques that are used to protect against decompilation and misuse.

0. Explain how the product precludes user modification of the encryption algorithms, key management and key space.

1. Check all that apply to this product:
Can the cryptographic functionality be easily changed by the user?
Yes ☐ No ☐

Does the product require substantial support for installation or use (beyond phone support, e.g. requiring a service contract).
Yes ☐ No ☐

Has the cryptographic functionality been modified or customized to customer specification.
Yes ☐ No ☐

2. You must certify that the application does not implement an open cryptographic interface (OCI). An OCI provides end users with the ability to plug in encryption functionality of their choice. Applications implementing an OCI are not eligible for download via the BREW platform.

☐ I certify that this software application does not contain an open cryptographic interface.

I hereby certify that the above information is correct.

1701

1702

1703

1. Terminology

This section describes terms and identifiers that will be used in this document. Table 1 defines cross-functional QIS Middleware terms and identifiers. This table is not meant to be a complete list of all QIS terms but specifically those terms that are relevant to the QDC applications and services.

Table 1. QIS Middleware Terminology

Name	Purpose
ISV	Independent Software Vendor. Term used to identify the developer company which has established a definitive agreement with QIS.
Restricted Application	A restricted application or "pass-thru" application is a non-certified BREW application submitted to QIS from the carrier. It is only available originating carrier for distribution.
Limited Application	A limited application is a BREW certified application which may be set to limited distribution to a select set of carrier(s) by the ISV.
Public Application	A public application is a BREW certified application that is made available to all carriers for distribution by the ISV.
Standard Application	Either a Limited or Public Application.
Appl ID	Assigned automatically by UAM and only relevant to UAM and ADS. Part Number Relationship. UAM maintains the mapping between part numbers and Appl IDs. Transaction History: Appl ID is contained in the transaction history and used to track phone events.
SID	Subscriber ID. This is a pseudo customer identifier that is used to identify customers. It may correspond to a MIN (i.e., phone number) or may be another carrier designated identifier. The SID and MIN will be logged at the ADS with the events will be provisioned on the phone. If a SID is provisioned on the phone, MIN will not be transferred back to TX.
MIN	Mobile ID Number or phone number The MIN is used for transaction processing if the Carrier elects not to provision a SID. In that case: Transaction History: MIN is passed in the transaction log for billing and payment processing. Provisioning Data Report: Carriers will provide a perioding report to QC which will include MIN updates (i.e., MIN out of service, MIN reassignment, etc.)
Part Number	▪ Is assigned by QDC Operations when the application is submitted for insertion into UAM. ▪ For Restricted Applications, the Carrier assigns a carrier unique part number and then QDC Operations adds a carrier prefix to the carrier part number. ▪ Part numbers are used for tracking the distribution and of BREW applications. It is a number exposed externally and a key identifier for billing process functions (i.e., not Appl IDs)
Primary Carrier	Associated to the carrier with which all BREW agreements are negotiated and approved. A primary carrier may be associated with "affiliate carriers" (i.e., for regional support).
DAP	Developer application price. This is the price which is used to determine the payment to issue the developer based on phone transactions. A developer may have different application DAP's across different carriers. It is independent of carrier's purchase price.
Purchase Price	The amount that the carrier will charge the device user.
Developer Fee (DF)	The amount that the developer is paid for the application.
QC Revenue Share (RS)	The amount that QC will charge the carriers based on application sales.
Pricing Methods	There are five pricing methods: 1) Demo 2) Purchase 3) Subscription 4) Upgrade (i.e., patch is a free upgrade) 5) Provisioned

Price Basis	There are four basis types associated with application pricing. 1) Fixed Uses. Billing based on 5 plays, 10 plays or unlimited. Developer/application is responsible for decrementing the counter. 2) Fixed Date. Billing based on expiration date ie: 3/31/00, date expressed as seconds since 0:0:0 6-Jan-1980 or unlimited 3) Fixed Duration. Billing based # of days after activation i.e., 30 days from first use or unlimited 4) Elapsed Time. Billing based on minutes of use ie: 120 minutes or unlimited
Price Point	For each price basis, one or more price points may be defined depending on the price method. The price point includes a value and a DAP. For example, for fixed uses a price point could be 5 uses and $1.00 DAP.
Price Plan	The price plan is a term used to refer to the entire pricing structure for a particular application/part number. The price plan includes all relevant price methods, basis types, and price points.

2. Application Pricing Template

The Independent Software Vendor (ISV) will submit application specific pricing information via the Developer Extranet. ISVs who submit their applications for TRUE BREW certification will be notified to submit application pricing information after their application has been certified. It is not mandatory for Restricted Applications to go through TRUE BREW certification, but they will be required to have pricing information submitted via the Carrier Extranet by the Carrier. This section describes the content and semantics of the pricing information.

2.1 Pricing Plans

a) Price plans ~~are~~ have both an effective date and expiration date~~d~~. This means that a current price plan may be superceded when a new price plan for the part number when the new plan reaches an effective date. Price locking agreements will impact timing of effectivity for new price plans. Effective dates for price plans will be validated against price locking rules using the expiration date of the active price plan.

b) Per Primary Carrier, only one plan may be in effect at any point in time (carrier specific or default) per part number~~.~~

c) Across Primary Carriers, a default price plan may be in effect across multiple primary carriers at any point in time.

d) ISVs will be able to set price locking attributes on a price plan.

e) ISVs will participate in price plan "negotiation" with carriers via the developer and carrier extranet services. The price plan negotiations may cause ISV to create carrier specific price plans or modify default price plan ~~for all~~across multiple carriers.

2.2 Pricing Methods

The developer must select at least one of the following application pricing methods:

1) Demonstration – is a method associated with no cost to the consumer.

2) Purchase – is a method associated with usage-based cost

3) Subscription – is a method associated with a monthly cost

4) Upgrade – is a method associated with an optional cost for application enhancements. An upgrade is a one-time enhancement purchase. It does not change the consumer's usage settings on the application being upgraded.

5) Provisioned – is a method associated with cost for preinstalled applications on a phone by the OEM and/or carrier. The developer will not have access to submit this pricing method unless the carrier or the OEM requests access on behalf of the developer.

2.3 Patches

Patches will be distributed with pricing method of type "upgrade" at no cost.

2.4 Pricing Basis

There are four valid pricing bases:

1) Number of Uses ~~Fixed Uses~~ (Number of uses as defined by the application)

2) Expiration Date (Fixed Date ~~(~~Day/Time GMT)

3) Number of Days (Fixed Duration) ~~(# Days)~~

4) Elapsed Time Used (minutes)

The pricing method dictates which pricing bases are applicable in the Pricing Template. The table below indicates which pricing bases correspond to each pricing method.

Pricing Method to Pricing Basis Mapping

Pricing Method	**Pricing Basis Types**			
	~~Fixed~~Number of Uses	**~~Fixed~~Expiration Date**	**~~Fixed Duration~~Number of Days**	**Elapsed Time**
Demonstration	X	X	X	X
Purchase	X	X	X	X
Subscription	~~N/A~~X	~~N/A~~X	~~N/A~~X	~~N/A~~X
Upgrade	N/A	N/A	N/A	N/A
Provisioned	X	X	X	X

2.5 One Pricing Basis per Pricing Method

For a given application/part number, a pricing method, with the exception of an upgrade, can be associated to only one pricing basis per price plan. For example, Price Plan A can have a Demonstration based on Fixed Date and a Purchase based on Fixed Duration. Price Plan B can have Demonstration based on Fixed Duration and Purchase based on Fixed Date. However, a single pricing method within a price plan cannot be both fixed duration and fixed uses.

2.6 Pricing Points

a) The purchase price method can have one basis with up to three price points defined.

b) The demonstration and provisioned price methods can have one basis with one price point defined.

c) Subscriptions will always be configured for unlimited use and are ~~not~~ associated to a specific pricing basis based on application considerations. ~~Therefore a~~A subscription will only have one price point. The price point will represent a monthly DAP and a monthly fee to the subscriber.

d) Upgrades are not associated with a pricing basis as an upgrade does not affect application usage of the version being upgraded. The only pricing information associated with an upgrade is an optional DAP and optional purchase price.

For example, an application of pricing method "purchase" can have a pricing basis of "Number of ~~fixed~~ uses" with price points 5 plays, 10 plays, and unlimited plays. Each price point is also associated with a DAP (Developer Application Price), which is one of the key values that the developer payment is derived from. Purchase price is not part of the pricing template as it is the value that the carrier will charge the consumer for that particular price point. Purchase price is maintained by the carrier via the Carrier Extranet catalog management functions. Purchase price can be set to be the same as, higher than, or lower than DAP. DAP is always in US Dollars, while Purchase Price currency is determined by the carrier.

Example Price Points for Number of Uses

Pricing Method = Purchase

Pricing Basis = Fixed Uses

Yen Conversion Example: 1 U.S. to 116 Yen

Attributes	Price Point 1	Price Point 2	Price Point 3
Value	5 plays	10 plays	Unlimited Plays
DAP	$1.00 U.S.	$3.00 U.S.	$100.00 U.S.
Purchase Price*	120 Yen	360 Yen	1,200 Yen

** Purchase Price is specified by the carrier and in the currency of the catalog*

2.7 Developer Default Price Plan

a) A price plan refers to the application price methods, pricing basis and price points (i.e., including DAP).

b) The ISV must define at least one price plan per application part number. (i.e., pricing method, pricing basis, and price points).

c) The ISV can define default and/or carrier specific price plans. Default price plans are visible to all carriers.

d) If the ISV did not define a carrier specific price plan, the carrier can add the application to their catalog using the default pricing plan.

e) Only the ISV can modify the application's price plan (i.e., by part number)

f) The carrier may attempt to negotiate with the ISV regarding price plans via the Carrier Extranet. If the ISV agrees to the modification, the ISV can modify their default price plan, if it is not in use with other carriers. More likely, an ISV will create a carrier specific price plan with the negotiated modifications.

2.8 Carrier Specific Price Plan

a) The ISV will be permitted to maintain carrier specific pricing plans. This means that the ISV can define distinct pricing methods, pricing basis, and price points per carrier.

b) Carrier specific price plans are only visible to the intended specific carrier.

c) The carrier specific pricing plan will replace the carrier pricing for the application in its entirety, (i.e., deltas between the default price plan and carrier price plan will not be retained).

2.9 Price Plan Effective dates

a) Price plans are effective dated. This means that a current price plan may be superceded when a new price plan for the part number reaches an effective date. The ISV can specify a price plan lock feature when preparing their pricing plans.

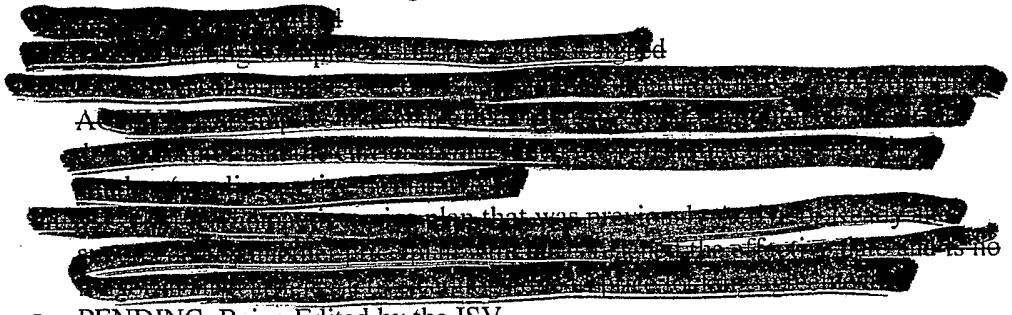
b) This price locking feature will apply to both the "default" price plans and "carrier specific" price plans.

c) Price Locking values will be No Lock, 6 Months and 12 Months:

- **No Lock**: means that the price can be changed for an active or pending application, with 60 days advanced notice to the carrier(s) from the day the price plan attributes are modified thru the extranet. An active application is one that is currently in a catalog offered to the carrier's consumers. A pending application is included in a catalog set to a Ready state (i.e., scheduled to be released). For non-active applications, new pricing will take effect based on effective date set by ISV. The Web interface will validate that the effective date of the price plan is a minimum of 60 days from the current date for active applications.
- **6 Months**: means that from the date the price plan is effective, the price plan cannot be changed by the developers within the six month window. The Web interface will validate that a new price plan for a part number with a six month lock must have an effective date beyond the six month lock date. The developer must make a new price plan effective at least 60 days before the end of the six months for it to be effective at the end of the current lock date. Any changes made within 60 days of the end lock date cannot go into effect for 60 days from the current date for active and/or pending applications. If no new plan is in effect at the end of the lock date, the current price plan remains in effect and price plan edits ~~ing~~ automatically rolls into a No Lock Plan.
- **12 Months**: same as 6-months but for 12 months.

d) The default price lock value will be No Lock.

e) Price Plans will have the following states:



- PENDING: Being Edited by the ISV

- READY: ISV Editing Complete, available for carriers to view. Effective Date Assigned. can't be edited until state is changed back to pending . Only ISV can change from ready to pending)
- ACTIVE: When price plan is assigned to any carrier parts list. A price plan in an ACTIVE state implies that is in effect for a specified part number/carrier. It does not infer that the carrier has included the part number/price plan in a catalog (pending, active or inactive)
- DEACTIVE: A price plan that was previously ACTIVE and superceded by another price plan which has reached the effective date. Also DEACTIVE if expired and not on any carrier parts list.

f) [redacted] When a price plan is set to READY, an e-mail notification should be generated to carrier marketing contacts for all carriers that have the related part number contained in PENDING or ACTIVE catalogs.

~~f)~~

g) When the effective date is reached, the price plan is in "effect" ~~across all catalogs~~. It is the carriers responsibility to update the carrier catalog with the new price plan and push the new catalog to the carrier ADS. If the part number is not included in any catalogs, the price plan still goes into "effect" on the effective date and not prior to it. It is the responsibility of the carrier to update the purchase price as appropriate in the catalog and push another catalog out.

- Option A: Carrier agrees with the new price plan and needs to modify their list prices accordingly on the effective date. If the ISV only changed DAP and there was no change in price points or bases, then new pricing will automatically go into effect for all existing catalogs that include the part on the effective date.
- ~~-~~Option B: There was a change in price points and/or bases and the carrier does not accept new pricing and update active catalog(s) [redacted] Carrier must remove the part number from all ~~pending~~ PENDING and ~~active~~ ACTIVE catalogs on effective date. [redacted] The middleware will monitor for carrier continued usage of expired price plans and notify ISV and Carrier.

3. Price Plan Functions by State

State	Editable	Clonable	Modify Effective Date	Delete Plan
~~Pending~~PENDING	X	X	X	X
~~Ready~~READY		X	X*	X**
~~Active~~ACTIVE		X		
~~Inactive~~DEACTIVE		X		

* Can only modify the effective date of a READY price plan (i.e., scheduled to be activated). Changes to the effective date of a READY price plan follow the same validation logic as above. For No Lock, the effective date must be 60 days from the current date.

** A READY price plan can be deleted because it has not reached its effective date.

3.1 Demonstration Applications

a) For Phase 1, a demonstration copy is the same application (i.e., Appl ID and Part Number) as the purchased copy (i.e., no difference in functionality).

b) For Phase 1, the demonstration copies will be configured for constrained "use" by intelligent price point.

c) Demonstration pricing may only have one price point per application.

d) The basis type selected for the demonstration copy does not have to correlate with the basis selected for application purchase.

e) Demonstration pricing will have a $0.00 DAP.

3.2 Subscriptions

In Phase 1, the goal is to implement a simple subscription model which can be enhanced to become more sophisticated in future phases. A basic scenario for subscriptions is described in Table 3:

Table 3. Subscription Scenario

Step	Description
1.	Application is made available for subscription to consumers via the Carrier Catalog Management functions in the Carrier Extranet and configures usage attributes
2.	Consumer downloads an application and selects price method of type subscription ~~(pre-paid or not)~~. A download event (DL) is recorded at ADS and propagated to QDC TX. TX converts this transaction into a "subscription start" event (SS) [redacted] on the same day as the DL event..
3.	Consumer uses application and is billed monthly for application usage.
4.	Consumer cancels subscription via the phone menu options.
5.	ADS logs the subscription end event (SE) with the subscription plan id. TX derives a delete event (DE) upon receipt of an SE event. The subscription end event (SE) is handed to billing to terminate the developer payment and consumer billing.
6.	Carrier applies subscription adjustments to the Adjusted Billing data based on the carrier's reconciliation process
7.	Adjustments affect QC invoicing according to carrier agreements.

The requirements to support the described subscription scenario are the following:

a) One Application. A subscription is a price method of an application. For Phase 1, subscriptions ARE NOT multiple applications under a common subscription plan.

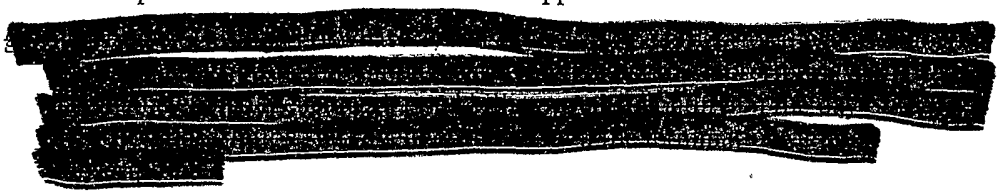
b) Monthly ~~Billing Subscription~~Subscription Bill Event. The TX manager will generate a monthly billing event for all subscriptions that have not been cancelled or remain "active" subscriptions. TX will generate the first subscription bill event at the start of the second month. As such if there is a subscription end event (SE) then a subscription start event (SS) in same month we will not double charge for that month. The derived billing date will be generated on the same calendar day per month that the application was downloaded for subscription. If the initial subscription download day is the 28, 29, 30 or 31 the monthly bill date will be on the 28th of each month.

c) Billing. A subscription will be billed to the consumer after application download. Discrepancies in subscription billing to the consumer may be adjusted using the carrier extranet services relating to billing adjustments.

d) Sustaining Content. A subscription may be associated with an application that requires on-going content updates (i.e., MP3's, Location Services, etc.). These content updates are to be provided by the developer or third parties. It is the responsibility of the carrier or ISV to initiate and stop the forwarding of content updates to the consumer based on the content subscription terms

e) Canceling a Subscription. A consumer cancels a subscription via a phone menu option. The subscription end event (SE) will get logged in the ADS and propagate to TX/Billing thereafter. TX derives a delete event (DE) upon receipt of an SE event. Permanently deleting the application is equivalent to canceling a subscription.

f) [redacted]/Disable Application. The user may ~~delete~~ disable the application without canceling the subscription. The consumer can subsequently download the application again for use. ~~We recommend using Disable over Delete to ensure no loss of personal data associated with the application.~~



3.3 Provisioned Applications

a) Provisioned applications are pre-installed on the phone by carriers and/or OEMs.

b) A provisioned application may be a restricted or standard, ~~limited or public~~ application.

c) A provisioned application will not be downloaded initially via the ADS, but the device users would have the ability to upgrade the application, restore the application after a disable, or extend the purchase through the ADS/Catalog.

d) For standard applications, the carrier must authorize the ISV to create a provisioned price plan. Either a carrier or OEM can submit a restricted application and price plan for provisioning through their respective extranets.

e) Provisioned price plan can use any price basis, but can only have one price point.

f) There is no purchase price associated to a provisioned application.

g) The license information is available to the end user via Mobileshop.

h) The device user can remove and/or disable provisioned applications from their phone via MobileShop.

6.2 Carrier Extranet

The carrier extranet will provide access to QDC services to authorized carriers. A user will be restricted to transact and view carrier specific information, where applicable. Some of the information on the carrier extranet is carrier specific and therefore restricted across carriers. Other information is general BREW information accessible to all carriers. A developer can create applications that will be restricted/limited for distribution by specific carriers. Other applications will be available for any carrier to distribute.

The types of extranet services include: authentication, catalog management, reporting, queries, customer service and secure file exchange. Table 7 describes the functionality in each of these areas.

Table 7. BREW Phase 1 Carrier Extranet Functions

Type	Functions	Sub-Functions
Authentication	Login	• Authenticate userid, password, token based
Catalog Mgmt	Manage Carrier Application List	• Manage carrier application list • Handle global restrictions based on platform, language and developer. • Query available applications by application type, developer, language, platform • Hide applications from the select list • Retrieve hidden applications • Add applications to carrier parts list. • Denote applications not yet included in catalog with special symbol • View application details read-only information (i.e., description of application, date of certification, developer information, etc.)
	Negotiate pricing w/developer	• Agree to developer pricing • Send email to developer with pricing modification
	Create Catalogs	• Create, edit and delete new catalogs • Set catalog name, version, description, effective dates, languages, currency • Clone and edit all states of existing catalogs • Manage catalog states: pending, ready and active • Ready, Active and Deactivated catalogs may not be edited – they can only be cloned.
	Manage Catalog categories	• Create, edit and delete categories • Associate icons to categories • Modify sort order of categories • Enter additional language for categories
	Manage Applications within Catalog Categories	• Modify sort ordering of Applications within Categories (i.e., Move function) • Add, Delete Applications • Within a single category, an application can have one price basis and pricing information for each pricing method. • Assign application level icons to be displayed on the phone. • The developer sets application names, but the carrier can append them. • An application can be named as differently across categories within a catalog
	Maintain Application Pricing	• List ALL applications available in Carrier Applications List • View Application Pricing Method and Basis Information • Within catalogs, the carrier can set the purchase price for each available pricing method. (i.e., displayed on the phone and can be different than Developer DAP). • The purchase price should be entered in the default currency of the catalog. • If the catalog supports multiple currencies, the purchase price must also be set in every currency the catalog supports. • The carrier can select specific pricing methods and specific pricing basis from the developer's pricing plan to host in a catalog.

281

Table 7. BREW Phase 1 Carrier Extranet Functions (continued)

Type	Functions	Sub-Functions
	Changing available applications for end user.	• If a new catalog is pushed out to an ADS and a previously offered application is not offered any more, the application may still be used by subscribers who downloaded it previously. If a user had disabled the application and it is no longer available via the current catalog, it is still available on the ADS to be reloaded on that subscribers phone.
	Associate catalog to ADS'	• Specify carrier ADS to activate with this catalog. A catalog/version can be associated to multiple carrier ADS'.
	Catalog Version Activation Date	• Propagates all changes to the selected catalog version to the ADS at specified Activation date/time.
Catalog Reports	Catalog Report	• Query all states of catalogs (pending, ready , active and deactivated) • Select Catalog and Version and see contents.
	BREW Application Cross Reference Report	• Listing of all BREW Application and which are active on any Carrier's ADS. List by Carrier ADS ID.
	Activation Report	• Listing ordered by date descending, of all changes submitted for a Test catalog or Prod catalog for a specified ADS ID. Should list versions of catalog applied over a date range and the changes to the catalog applied at the activation date (i.e., application add/delete, pricing change, userid submitting the change, etc.).
Carrier Guidelines	Maintain	• A carrier maintains and publishes Carrier specific guidelines which are used to inform developers about carrier specific application attributes and pricing information.
Developer Info	Query for Developer Information	• Query developer contact information by Part Number, Developer Name or Application Name
Application Recall	Application Recall List	• Enter Application Delete List for application recall/delete across all Carrier ADS'
Provisioned Application	Allow Developer to Submit Provisioned Price Plan	• Mark application for provisioned pricing • Display provisioned pricing • Remove provisioned pricing from application
Restricted Application	Submit	• Submit Application • Create Price Plans • Modify Price Plans
Billing Services	Billing Inquires	• Near real-time queries
	Billing Event Adjustments	• Enter SID, Part Number and Date Range • Retrieve events associated with parameters • Mark event as "ignore" or enter event adjustment debit/credits
Secure File Exchange	QDC to Carrier	• Monthly Billing Related – Adjusted Billing Data Report – includes SID, QC Part Number, Application Title, Date and time of download in GMT, basis for purchase, and price of application. This report distinguishes between applications downloaded and applications deleted by the subscriber. It includes processing of all the following types of data at a snapshot in time: a) *Transaction History* from the Carrier ADS (includes part number conversion) b) *Carrier Provisioning Data* which includes Preinstalled Applications via OEMs of Carrier Service Centers, MIN Deactivation/Reassignment c) *Adjustments* – QC adjustments, carrier credits and accounting reconciliations for subscriptions • Application Recall Data – used by carrier to broadcast an SMS messages to the phones to recall malicious or volatile applications • Application Technical Document – used by carrier for tier 1 support to consumers.
	Carrier to QDC	• Monthly Billing Related – Carrier Provisioning Data which includes: a) *Preinstalled Application data* – includes SID, Application Title, QC Part Number, Installation Date/Time, and Price Plan b) *MIN Deactivation/Reassign data* – MIN, and date/time of activation, need to know if MIN history needs to be carried over from previous number to new number c) *Bulk Adjustments data* – by exception there may be file transfers instead of entering the adjustment via the extranet one transaction at a time, This is TBD. Customer Service Related - BREW Problem Summary Report - Monthly Summary of all BREW Related Call and Problems

282

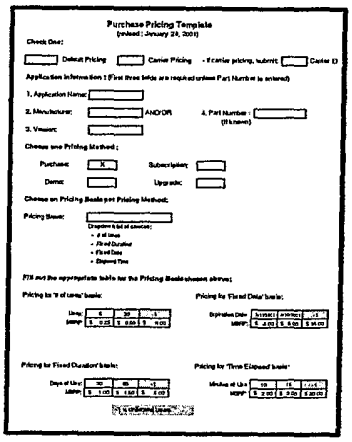
282 ***Price Plan Overview***

283

Developers Submits Pricing Templates

Five Pricing Methods:

1. **Purchase**
2. **Upgrade**
3. **Demonstration**
4. **Subscription**
5. **Provisioned**



* *Purchase Price is specified by the carrier*

QUALCOMM Confidential May 2001 QUALCOMM

284

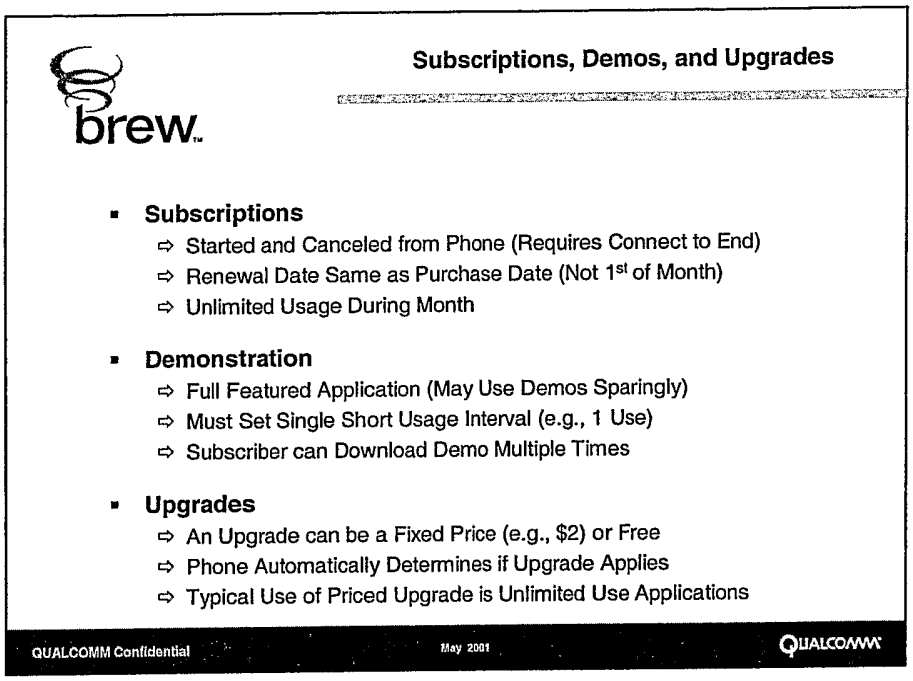
Developer Pricing Options

Developer has Choice of:

- **Fixed Number of Uses [5 Application Launches]**
  - ⇨ Developer Defines a Use (Decrement/Increment)
  - ⇨ Example: Off-Line Game
- **Fixed Date [June 5, 2001]**
  - ⇨ Example Use: Limited Time Special Pricing
- **Elapsed Time Used [20 Minutes]**
  - ⇨ Actual Time Application in Use
  - ⇨ Example Use: Group On-Line Gaming
- **Fixed Duration [90 Days]**
  - ⇨ Elapsed Calendar Time on Phone
  - ⇨ Example: Off-Line Music Player

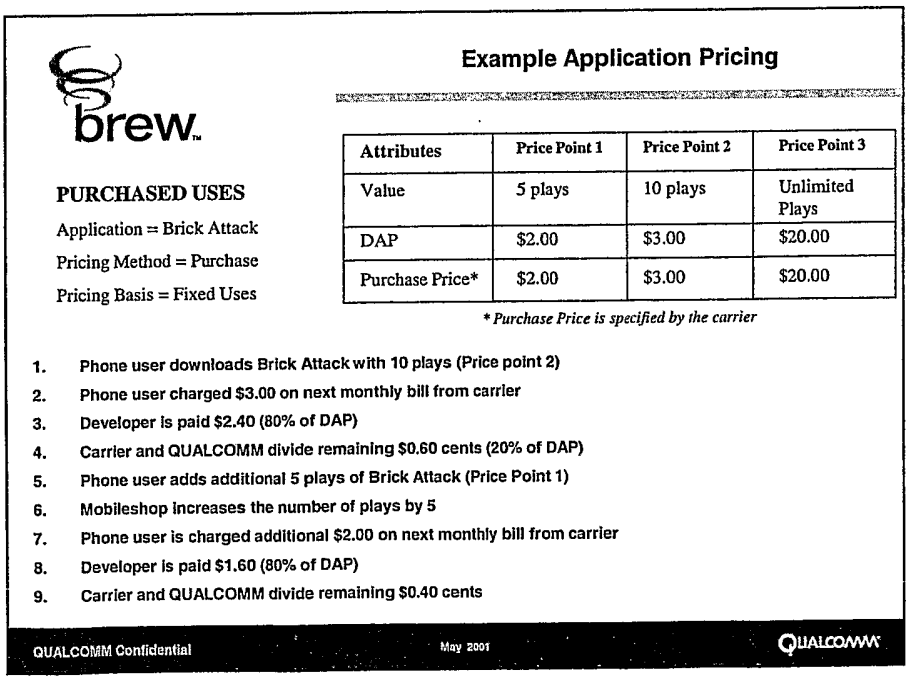
QUALCOMM Confidential May 2001 QUALCOMM

285



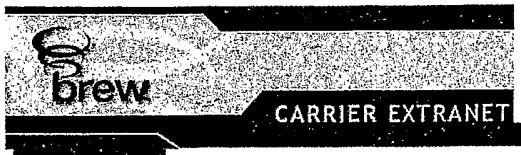
286

287



288

289



Welcome Mitch Oliver, Qualcomm LOGOUT

QUALCOMM

Thu, August 9, 2001 (GMT+8)

- HOME
- APP LIST
- ADD NEW CATALOG
- CATALOGS
- ADS ADMIN
- PARTS LIST
- REPORTS
- HELP

APPLICATIONS LIST

Hangman - Application Details

Name:	Hangman
Description:	Hangman signed application
Version:	1.1
Developer:	WART
Type:	Games
Languages Supported:	English

Price Plan:

Summer Specials- Effective Date: 07-21-2001

Demo

Uses	List Value: 5	DAP: $0

Purchase

Uses	List Value: 25	DAP: $2.25
Uses	List Value: 50	DAP: $3

Agree to Price Plan

To use Hangman in a Qualcomm catalog, click on the "I Agree" button to agree to the price plan. By agreeing to the price plan, this application will now be available for you to add to a catalog.

I Agree

Request Price Change

If you would like WART to modify the price plan for Hangman before selecting the application, type in what the requested modification is and click Send.

Requested Change:

Carrier [redacted] requests a change in Purchase 25 uses DAP from $2.25 to $2.00

Send

Maintained by brew-web@qualcomm.com. Version 1.1
(moliver,tst,Qualcomm)

Welcome Mitch Oliver, Qualcomm LOGOUT

Thu, August 9, 2001 (GMT+8)

brew CARRIER EXTRANET

- HOME
- APP LIST
- ADD NEW CATALOG
- CATALOGS
- ADS ADMIN
- PARTS LIST
- REPORTS
- HELP

APPLICATIONS LIST

Hangman - Application Details

Name:	Hangman
Description:	Hangman signed application
Version:	1.1
Developer:	WART
Type:	Games
Languages Supported:	English

Price Plan:

Summer Specials- Effective Date: 07-21-2001

Demo			
	Uses	List Value: 5	DAP: $0
Purchase			
	Uses	List Value: 25	DAP: $2.25
	Uses	List Value: 50	DAP: $3

Agree to Price Plan

To use Hangman in a Qualcomm catalog, click on the "I Agree" button to agree to the price plan. By agreeing to the price plan, this application will now be available for you to add to a catalog.

I Agree

Request Price Change

If you would like WART to modify the price plan for Hangman before selecting the application, type in what the requested modification is and click Send.

Requested Change:

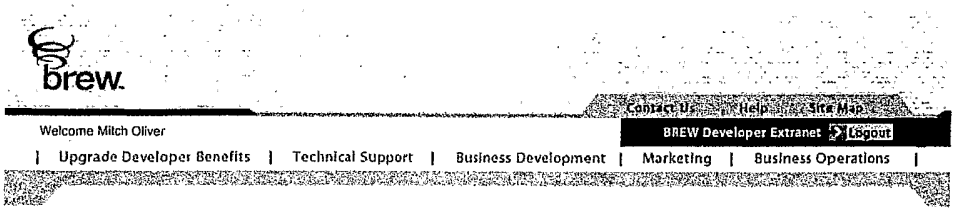
Send

Price Plan Change Request History

History of price request changes that have been submitted for Hangman by Qualcomm

- Verizon requests a change in Purchase 25 uses DAP from $2.25 to $2.00 (2001 -08-10 00:02:10.0)

Maintained by brew-web@qualcomm.com. Version 1.1
(moliver,tst,Qualcomm)



Pricing Templates

Create New Price Plan

Pricing Type

(•) Default Pricing

( ) Carrier Specific

Price Plan Information

Price Plan Name: [ ]

There is no effective date for inital pricing of an application. However, a developer can set an effective date for price point changes once the application pricing has been established. The carrier has up to 60 days to either accept the revised pricing or remove the application from their catalog.

The price changes indicated in this template will be effective on: [ ]
(The date must be formatted DD-MMM-YYYY hh:mi:ss.)

Price Plan Effective Period: [No Price Lock]

Demo Basis

Developer must select one price basis for the demo price method. Developer Application Price (DAP) must and will always equal zero dollars for demo applications. This field is not updateable by the developer or the carrier.

Choose one:
[(no demo)]

Purchase Pricing Basis

Developer must select one pricing basis for the purchase price method. Within this chosen basis, the developer may set up to three price points. One of the price points can be designated for unlimited uses. Developer Application Price (DAP) must be set for each price point in the U.S. Dollars.

Choose one:
[(no purchase plan)]

Subscription Basis

Pricing for the subscription price method is indicated as a monthly charge for unlimited uses within that month. Developer Application Price (DAP) must be sent in U.S. Dollars.

Choose one:
[(add subscription plan)]

Upgrade Basis

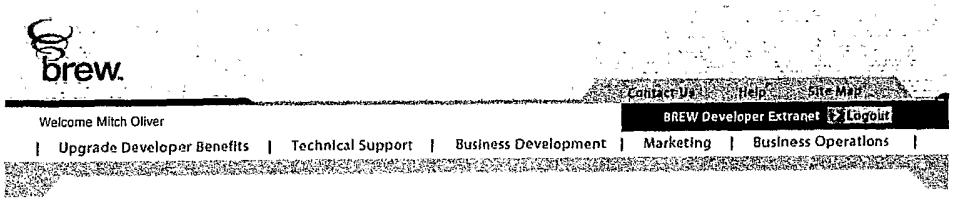
The developer has the option to set a one time fee to the end user for downloading the upgrade. If this fee is not set, the application will be placed in the carrier's catalog as a free upgrade. For this price method, Developer Application Price (DAP) indicates the amount of this one time fee and must be set in U.S. Dollars.

Choose one:
[(add upgrade)]

[Submit] [Reset] [Cancel]

QUALCOMM

Site last updated: August 8, 2001



Pricing Templates

Create New Price Plan

Pricing Type

(•) Default Pricing

( ) Carrier Specific

Price Plan Information

Price Plan Name: PP2

There is no effective date for inital pricing of an application. However, a developer can set an effective date for price point changes once the application pricing has been established. The carrier has up to 60 days to either accept the revised pricing or remove the application from their catalog.

The price changes indicated in this template will be effective on: 30-aug-2001 00:00:00
(The date must be formatted DD-MMM-YYYY hh:mi:ss.)

Price Plan Effective Period: No Price Lock

Demo Basis

Developer must select one price basis for the demo price method. Developer Application Price (DAP) must and will always equal zero dollars for demo applications. This field is not updateable by the developer or the carrier.

Choose one: Expiration Date

Enter expiration date and price below:

Expiration Date: 31-dec-2001 00:00:00

DAP (U.S. Dollars): $0.00

Purchase Pricing Basis

Developer must select one pricing basis for the purchase price method. Within this chosen basis, the developer may set up to three price points. One of the price points can be designated for unlimited uses. Developer Application Price (DAP) must be set for each price point in the U.S. Dollars.

Choose one: Expiration Date

Enter expiration dates and prices below (dd-mmm-yy):

Expiration Date:	31-dec-2001 00:00:00	31-oct-2001 07:00:00	(•) Unlimited ( ) (set a value)
DAP (U.S. Dollars):	$5.00	$3.00	$

Subscription Basis

Pricing for the subscription price method is indicated as a monthly charge for unlimited uses within that month. Developer Application Price (DAP) must be sent in U.S. Dollars.

Choose one: Expiration Date

Enter price below:

Subscription Plan: monthly

DAP (U.S. Dollars): $ 1.00

Upgrade Basis

The developer has the option to set a one time fee to the end user for downloading the upgrade. If this fee is not set, the application will be placed in the carrier's catalog as a free upgrade. For this price method, Developer Application Price (DAP) indicates the amount of this one time fee and must be set in U.S. Dollars.

Choose one: (add upgrade)

Submit | Reset | Cancel

QUALCOMM

Site last updated: August 8, 2001

brew CARRIER EXTRANET

Welcome [redacted]

QUALCOMM

Thu, August 9, 2001 (GMT+8)

- HOME
- APP LIST
- ADD NEW CATALOG
- CATALOGS
- ADS ADMIN
- PARTS LIST
- REPORTS
- HELP

APPLICATIONS LIST

Hangman - Application Details

Name:	Hangman
Description:	Hangman signed application
Version:	1.1
Developer:	WART
Type:	Games
Languages Supported:	English

Price Plan:

Summer Specials- Effective Date: 07-21-2001

Demo	Uses	List Value: 5	DAP: $0
Purchase	Uses	List Value: 25	DAP: $2.25
	Uses	List Value: 50	DAP: $3

Agree to Price Plan

To use Hangman in a Qualcomm catalog, click on the "I Agree" button to agree to the price plan. By agreeing to the price plan, this application will now be available for you to add to a catalog.

I Agree

Request Price Change

If you would like WART to modify the price plan for Hangman before selecting the application, type in what the requested modification is and click Send.

Requested Change:

Carrier [redacted] requests a change in Purchase 25 uses DAP from $2.25 to $2.00

Send

Maintained by brew-web@qualcomm.com. Version 1.1
(moliver,tst,Qualcomm)

CLAIMS

1. A system for distributing applications over a wireless network, comprising:

a centralized processor operable to perform administrative functions associated with downloading an application to a wireless device;

a local processor connected to the centralized processor and operable to receive catalog data and the application from the centralized processor, to transmit the catalog and application to the wireless device, and to record transaction data associated with the transmission of the application; and

a transaction history server connected to the centralized processor and the local processor and operable to receive metadata information from the centralized processor, receive transaction data from the local processor, and processor the metadata information and transaction data for billing.

2. An apparatus as disclosed herein.

3. A method as disclosed herein.