Chapter 19

Providing Prepaid Services

Prepaid service applications can be easily integrated with the SRC software. We provide the prepaid services demonstration application (referred to hereafter as the prepaid services demo) to illustrate the concept.

This chapter contains the following sections:

Overview of Prepaid Services Demo

Prepaid service applications assume that users (subscribers) are registered with a subscriber management system, so that they can log in to the network and be authenticated. By default a subscriber has no access to the network. All attempts to access the Internet are intercepted and captured by the Service Selection Portal.

Subscribers pay for the service in advance of use by purchasing an access card that has a valid account number and expiration date. The subscriber then can activate the prepaid service, which might be, for example, access to the Internet or access to a gaming server. At activation, the portal prompts the subscriber for the account number and validates the access. The portal grants access if appropriate, and charging starts as soon as the portal grants access.

To integrate prepaid service applications, the prepaid services demo consists of the following two components:

- The prepaid account server is a Solaris package, UMCppdemo, located in the *solaris* directory on the SRC application library CD.
- The Prepaid Account Administration application is a standard Web application archive (WAR file) located on the SRC application library CD, */webapp/accountAdmin.war*. You must install this component in your application server, such as JBoss, and configure the object reference for the account server.

The prepaid services demo supports two types of prepaid service applications, time based and volume based. Both types are limited in the demo to a single service being concurrently active per prepaid account. The account server maintains the accounts.

Account Server

The account server is the central data repository for the prepaid services demo. It maintains the different accounts and provides access for the other SRC components. The account server is a CORBA server with a data storage backend. In the prepaid services demo, data is stored on the local file system; in a real application you should use a relational database management system (RDBMS) for data storage.

The account server employs the SAE plug-in interface. The server publishes an object reference to a standard COS naming service or to a file in the local file system. It uses the managed accounts to authorize access to prepaid services and updates the accounts based on actual usage.

The model assumes that subscribers can log in and be authenticated. By default, all attempts to access the Internet are intercepted and captured by the portal. Subscribers pay for the service in advance of use by purchasing a valid access card. The subscriber then can activate the prepaid service. At activation, the portal prompts the subscriber for the account number and validates the access. The portal grants access if the account exists, has not expired, is not locked, and has a remaining balance (time or volume) greater than 0. When it grants access in response to a request, the account server locks the account against concurrent access. Charging starts as soon as the portal grants access.

Time-Based Services

Time-based services are sold by access time. These services have no limits placed on the data transmitted. For example, voice long distance service accounts are measured in connection time.

When it authorizes a time-based prepaid account, the account server sets the session timeout based on the current balance of the account. The account server locks the account when the session start is signaled. When the session stop is signaled, the account server updates the account based on the session time and unlocks it.

When the service stops (because the subscriber stops service on the portal, the subscriber logs out, or the session timeout expires), the account is unlocked, and its time balance is decreased by the session time.

Volume-Based Services

Volume-based services are sold based on upload or download data volume. For the demo application, volume is defined as the sum of the upload and download volumes. A real implementation might distinguish between the two for accounting purposes.

When it authorizes a volume-based prepaid service, the account server sets an interim update interval according to the following formula:

$$\text{interim update interval} = \frac{\text{remaining volume in account}}{\text{maximum bandwidth available to subscriber}}$$

The maximum bandwidth is the greater of the two plug-in attributes upstreamBandwidth and downstreamBandwidth. If you do not specify values for these attributes, then the maximum bandwidth defaults to 1 Mbps.

When the session start is signaled, the account server locks the account.

When an interim update is signaled, the account server updates the interim update interval based on the account balance and the current consumption. It compares the volume used so far in the session with the remaining volume. If the session volume is greater than the remaining volume, the account server sets the session timeout to zero to stop the session.

If the session volume is less than the remaining volume, the interim update interval is recalculated, and no further action takes place until the end of that interval.

The interim update interval must be larger than a specified minimum value; the demo application employs a minimum of 5 minutes. This feature enables accounts to be overdrawn by an amount equal to the maximum bandwidth times the minimum time.

When the session stop is signaled, the account server updates the account based on the volume counters and unlocks it.

Installing and Configuring the Prepaid Services Demo

You must install and configure both the account server and the Prepaid Account Administration application. Additionally, you must configure the prepaid plug-in on the SAE and create and configure the service(s) that will use the prepaid accounts.

Installing the Account Server

You must manually install the UMCppdemo package to deploy the account server.

```
pkgadd -d /cdrom/cdrom0/solaris UMCppdemo
```

NOTE: The prepaid services demo is provided on the SRC application library CD.

For information about installing the prepaid services demo, see *SRC Application Library Guide, Chapter 1, Installing the SRC Applications*.

Configuring the Account Server

Before you start the account server for the first time, you must run a script to configure it. To configure the account server:

1. On the SAE host, log in as `root` or as an authorized nonroot admin user.

2. Launch the configuration script from the prepaid services demo installation directory.

   /opt/UMC/prepaid/etc/config

3. The configuration script prompts you for input and confirms your choices, as in the following example:

```
Which naming prefix shall be used for publishing the objects?
[demo/accountServer] [?,q]
demo/accountServer
Which naming server do you want to use? [] [?,q]
corbaname::localhost
Which file name prefix shall be used for publishing the objects? [] [?,q]
/var/tmp/accountServer
Which user-id shall be running the account server? [nobody] [?,q]
nobody
COSName:    "demo/accountServer"
NameServer: "corbaname::localhost"
IORFile:    "/var/tmp/accountServer"
USERID:     "nobody"
Is this correct? y
```

4. The script configures the account server according to your responses.

Publishing the Object References

The sample configuration presented above configures the account server to publish the object references to a COS naming service and to a local file. Depending on your needs, you might want to choose only one or the other method.

NOTE: The account server and the Prepaid Account Administration application must run on the same host for the local file feature to work. If you install these components on multiple hosts, you must configure the account server to publish the object references to a COS naming service.

When you publish the objects to a COS naming service, you specify the prefix of the published name, such as demo/accountServer, and the URL of the name server, such as corbaname::localhost. In this case the account server publishes the object reference of the plug-in to the URL corbaname::localhost#demo/accountServer.plugin. The account server publishes the object reference of the account manager to the URL corbaname::localhost#demo/accountServer.acctMgr.

The local file is specified by the path and prefix of the filename, */var/tmp/accountServer*. In this case the account server publishes the object reference of the account manager to */var/tmp/accountServer.acctMg*r and the object reference of the prepaid plug-in to */var/tmp/accountServer.plugin*.

Manual Configuration

Although the configuration script is sufficient to configure the account server for most purposes, you can also configure the server by using the command line.

- To publish the object references into a local file, specify the path and prefix of the filename:

  ```
  #accountServer -f <fileNamePrefix>
  ```

- To publish the object references to a COS naming service, specify the prefix of the published name:

  ```
  #accountServer -c <namePrefix>
  ```

- The COS naming server is taken from the initial references. You can do one of the following:

  - Globally configure omniORB in the file */etc/omniORB.cfg*
  - Specify the following option when you configure the account server:

    ```
    -ORBInitRef NameService=corbaname::nameServerHostname
    ```

  For example, to publish the object references to a COS naming server running on server ns.domain.com, configure the account server as follows:

  ```
  #accountServer -ORBInitRef NameService=corbaname::ns.domain.com -c
  demo/accountServer
  ```

- If you start the account server as a root user, the account server switches the user ID to an unprivileged user after initialization. The default user ID is nobody. To override the default value, specify a different user:

  ```
  #accountServer -f /var/tmp/accountServer -u <username>
  ```

Starting the Account Server

To start the account server:

1. On the account server host, log in as `root` or as an authorized nonroot admin user.
2. Start the account server from the root directory.

   /etc/init.d/accountServer start

   The system responds with a start message.

Stopping the Account Server

To stop the account server:

1. On the account server host, log in as `root` or as an authorized nonroot admin user.

2. Stop the account server from the root directory.

   /etc/init.d/accountServer stop

   The system responds with a stop message.

Configuring the SAE for the Prepaid Plug-In

You configure the prepaid plug-in in the same way that you configure other SAE external plug-ins. For information about configuring SAE plug-ins, see:

- *SRC-PE Subscribers and Subscriptions Guide, Chapter 9, Configuring Internal, External, and Synchronization Plug-Ins with the SRC CLI*
- *SRC-PE Subscribers and Subscriptions Guide, Chapter 10, Overview of Configuring Plug-Ins for Solaris Platforms*

The properties for this plug-in are as follows.

Plugin.prepaid.objectref

- Specifies the reference of the plug-in object implemented by the account server.
- Value—Depends on the host and the configuration of the account server; that is, whether the object reference is published to a COS naming service or a local file
- Examples

  In the following example, the object reference has been published to a COS naming service running on the host ns.domain.com:

  Plugin.prepaid.objectref = corbaname::ns.domain.com#demo/accountServer.plugin

  In the following example, the object reference has been published to a local file on the host:

  Plugin.prepaid.objectref = file:/var/tmp/accountServer.plugin

Plugin.prepaid.attr

- Defines the attributes used by the plug-in.
- Value—Use only the following value:
  Plugin.prepaid.attr = PA_UID,PA_AUTH_USER_ID, PA_SESSION_TIME, PA_DOWNSTREAM_BANDWIDTH, PA_UPSTREAM_BANDWIDTH

Configuring the Prepaid Services

Each defined service that uses prepaid accounts must be configured to use the prepaid plug-in as its authorization and tracking plug-in. For example, suppose you have a GameMaster premium gaming service for which you want to use prepaid accounts. You must create this service with SDX Admin and enter the value "prepaid" into the Authorization Plugin and Tracking Plugin fields. See *SRC-PE Services and Policies Guide, Chapter 2, Managing Services on a Solaris Platform* for more information about creating and configuring services with SDX Admin.

Deploying the Prepaid Account Administration Application

You must deploy the WAR file for the Prepaid Account Administration application in the Web application server.You can find this file, *accountAdmin.war*, in the folder *webapp* on the SRC application library CD. Refer to the documentation for your Web application server for information about deploying applications.

For example, to deploy the Prepaid Account Administration application inside JBoss, copy the file to the JBoss */server/default/deploy* directory.

```
cp /cdrom/cdrom0/webapp/accountAdmin.war
/opt/UMC/jboss/server/default/deploy
```

JBoss automatically starts the application when a new WAR file is copied into the deploy directory.

Configuring the Prepaid Account Administration Application

You must configure the Prepaid Account Administration application with the object reference of the account manager. Configure the object reference as a <context-param> in the *WEB-INF/web.xml* file from the *accountAdmin.war* file. The parameter name is acctMgr, and the value is a CORBA URL of the account manager object reference, as in the following example:

```
<context-param>
  <param-name>acctMgr</param-name>
  <param-value>corbaname::ns.domain.com#demo/accountServer.acctMgr
</param-value>
</context-param>
```

Managing Prepaid Accounts

Use the Prepaid Account Administration application to manage prepaid accounts.

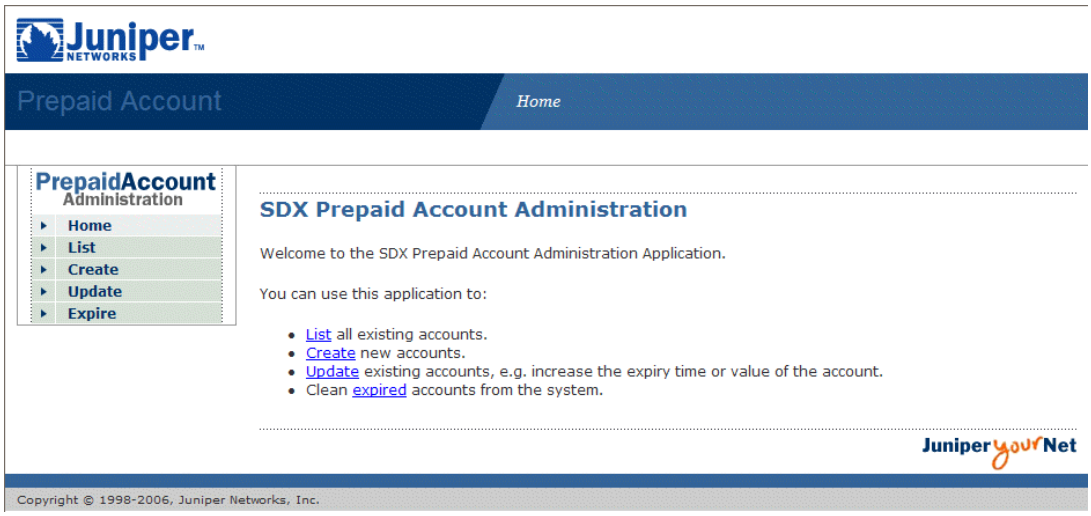
Accessing the Prepaid Account Administration Application

To access the Prepaid Account Administration application, enter the following in your Web browser:

http://<host>:8080/accountAdmin

where <host> is the name or IP address of the workstation on which you installed the Prepaid Account Administration application.

The Prepaid Account Home page appears.



Administering Accounts

On the Prepaid Account Home page, you can select to list, create, update, or clear accounts.

- Listing an Account—You can display summary information for accounts. You can also display the state of an account selected by its account number.

- Creating an Account—On the Create Accounts page, you can create multiple similar accounts simultaneously, but all must have the same type (time or volume), balance, and expiration (expiry) date.

  Specify the expiration date in the format YYYYMMDD. You can optionally specify an expiration time after the date in the format HHmm. If you do not specify a time, the account expires at midnight on the specified date. For example, to set the expiration for July 21, 2004 at 08:35 a.m., specify the following in the Expiry Date field:

  200407210835

  After completing the account fields, click OK to create the account(s).

- Updating an Account—On the Update Accounts page, you can manually credit an account, extend its expiration date, or unlock it.

- Clearing an Account—On the Clear Accounts page, you can delete expired accounts.