White Paper

radisys®

# CPE WAN & FAP Management Protocols

*By: Ravi Raj Bhat, Vice President of Engineering and V. Srinivasa Rao, Sr. Architect*

## Overview

Standardization of Femtocell technology has come a very long way. This progress is largely due to the leadership of the telecom industry consortium known as the Femto Forum (www.femtoforum.org) working with various standardization organizations such as the 3rd Generation Partnership Project (3GPP, www.3gpp.org) and the Broadband Forum (BBF, previously known as the DSL Forum, www.broadband-forum.org).

## CONTENTS

**BBF Auto-Configuration Architecture and Framework** *pg. 2*

**Auto-Configuration Data Organization** *pg. 2*

**Data Hierarchy** *pg. 3*

**Profiles** *pg. 3*

**CPE WAN Management Protocol Overview** *pg. 4*

**Protocol Stack and Operation** *pg. 4*

**Protocol Operation** *pg. 5*

**FAP Data Model** *pg. 5*

**Control Object Group** *pg. 6*

**Configuration Object Group** *pg. 7*

**Monitoring and Management Object Group** *pg. 7*

**References** *pg. 8*

3GPP has standardized the Fa interface, known as the Iuh interface in 3GPP terminology (identified by the Femto Forum reference model shown in Figure 1). 3GPP has made good progress in defining the HNBAP protocol for communication between a Femto Access Point (FAP, also known as a Home NodeB or HNB in 3GPP terminology) and a Femto Gateway (FGW) and is actively identifying a means of transporting Radio Access Network Application Part (RANAP) to the RNC (including definition of the RANAP User Adapatation, RUA). The Femto Forum also worked with BBF to standardize the Fm interface by leveraging the existing remote device configuration and management standards TR-069 and TR-106.

This paper focuses on explaining the existing BBF remote device management framework and how it would be adapted to manage FAPs. Technical references are provided at the end of the document.

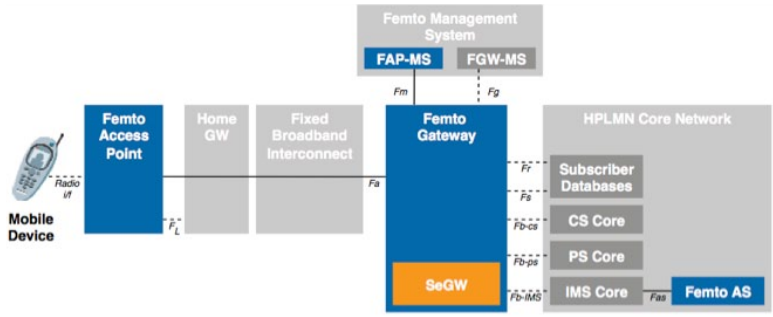


**Figure 1.** *Femtocell Reference Model (Source: Femto Forum)*

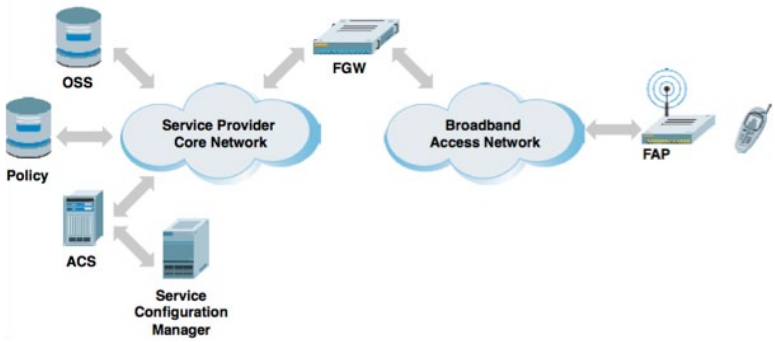


**Figure 2.** *Auto-Configuration Network Architecture*

## BBF Auto-Configuration Architecture and Framework

The guiding principle around BBF's auto-configuration architecture and framework is to configure the Customer Premises Equipment (CPE, also known as the Broadband Network Termination, or B-NT in BBF terminology) with "zero touch" based on a pre-defined service configuration template located somewhere in the service provider's network, thus avoiding a costly truck roll to the customer premises and enabling a true plug-and-play experience for the customer.

Figure 2 illustrates the end-to-end network architecture to achieve auto-configuration. The Auto- Configuration Server (ACS) stores the pre-defined service configuration template, added and pre- validated by the service provider through its service configuration manager. BBF's technical report, TR-046, defines the DSL auto-configuration architecture and framework. This generic framework can be divided into three distinct aspects:

- Data organization, validation, and storage, which is typically described in a data model for a specific technology (e.g., TR-106).
- Process and transport protocol used to convey the configuration information from the ACS to a CPE device; this is described in TR-069.
- Configuration information used in the CPE device—leading to service activation—which is device-specific.

The following sections will briefly explain data model organization and TR-069, the CPE WAN Management Protocol (CWMP), followed by the specific data model being proposed by the Femto Forum and BBF to manage FAPs.

## Auto-Configuration Data Organization

A very important aspect of auto-configuration and flow-through service provisioning is the way configuration and service information is represented in the service provider domain. Such information needs to enable service providers to easily extend the configuration representation and quickly define new services with minimal incremental costs to roll out the new services. Configuration information representation is referred to as Data Model and an object-oriented model is preferred for this purpose as it allows easy extensibility and fits very well with the underlying Remote Procedure Call (RPC) methods used by protocol-independent transport mechanisms. TR-106 describes the data model template for TR-069-enabled devices. The following sections explain this template briefly.

## Data Hierarchy

Data representation for a TR-069 capable device will always have a single Root object which will be called either Device or InternetGatewayDevice. Typically the Root object contains two types of sub-elements: the Common Objects, applicable only to a Device Root object, and a single Services object that contains all the Service objects associated with the specific services or applications. For InternetGatewayDevice, the Root object will also contain the application-specific objects associated with it. A single device might include more than one service object (e.g., a device that serves both as a FAP and an IPTV set top box might include both FAP-specific and IPTV-specific Service objects) and/or more than one instance of the same type of Service object (e.g., where a TR-069 capable device proxies the management functions for one or more other devices that are not TR-069 capable). Figure 3 illustrates the Data Hierarchy and how the template is defined. Figure 4 pictorially illustrates the TR-069 Data Model Structure where each box represents an object container.

Re-definition of the Service object or Root object over time is allowed via Object Versioning with the first version starting at "1.0". Object version is defined as a pair of integers ("major"."minor") separated by ".", where the first integer is the major version number and the second integer is the minor version number. The major version changes if the subsequent version is not compatible with the previous versions, otherwise only the minor version changes.

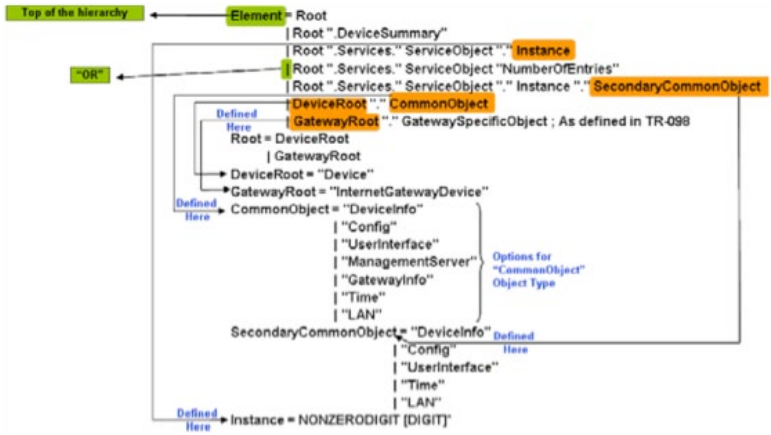


**Figure 3.** *Data Hierarchy*[3]

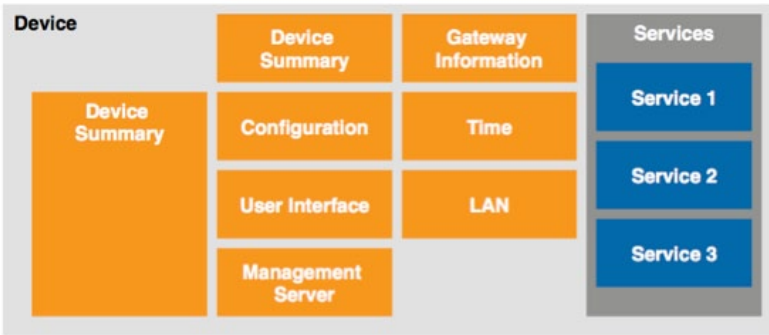


**Figure 4.** *TR-069 Data Model Structure*

## Profiles

The ACS needs to be scalable enough to communicate with multiple devices with varying capabilities and potentially from different manufacturers. This variability is controlled by defining the Profiles that express a collection of requirements associated with a given object, support for which can be explicitly indicated by the device. A device supporting a profile means that the device supports all of the requirements defined by that profile. The use of profiles allows the ACS a shorthand means of discovering support for entire collections of capabilities in a device.

A given profile is defined only in the context of a specific Service object or Root object with a specific major version. A Profile's name must be unique among profiles defined for the same object and major version. A given profile is defined in association with a minimum minor version of a given object that includes all of the required elements defined by the profile. For each profile definition, the specific minimum version must be explicitly specified.

For a given type of Service object, multiple profiles may be defined and they may have either independent or overlapping requirements. To allow the definition of a profile to change over time, the definition of every profile has an associated version number which uses a minor-only version numbering convention. All compatible changes to a profile use the same profile name but different minor versions. Any incompatible change to a profile shall use a different profile name. For every Service and Root object there is at least one Baseline profile defined which supports the minimum requirements required for any device that supports that object.

## CPE WAN Management Protocol Overview

TR-069 defines a CPE WAN Management Protocol (CWMP) for secure auto configuration of CPE devices and provides other CPE management functions in a common framework, including:

- Auto Configuration and Dynamic Service Provisioning of a CPE device either on initially connecting to the broadband network or later while re-provisioning or re-configuring to allow services and capabilities to change in the future. CPE devices are identified based on various criteria such as CPE vendor, model, software version, etc.
- Software/Firmware image management including mechanisms for version identification, file (group or single) download initiation (ACS initiated downloads and optionally CPE initiated downloads), authentication of file source, and notification of the ACS of the success or failure of file download.
- Status and Performance monitoring of the CPE device.

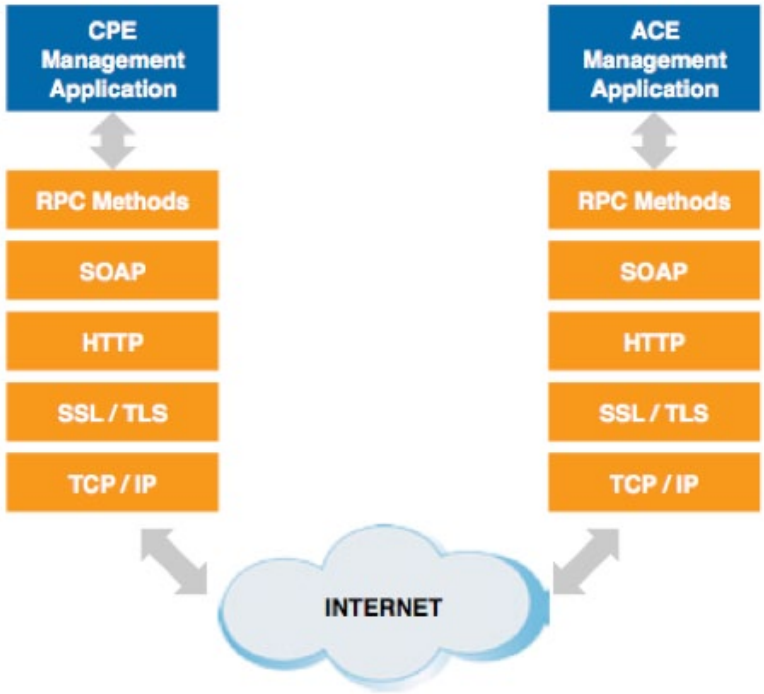


**Figure 5.** *TR-069 Protocol Stack*[1]

- Diagnostics for CPE to report critical information to the ACS, which may use it to diagnose and resolve connectivity or service issues as well as execute specific diagnostic tests.
- Security to prevent tampering with the management functions of a CPE or ACS, ensure confidentiality of the transactions that take place between them, allow appropriate authentication for each type of transaction, and prevent theft of service. CWMP to a large extent leverages the security services provided by underlying layers (e.g., SSL/TLS).

## Protocol Stack and Operation

Figure 5 illustrates the protocol stack used in the CWMP. CPE/ACS Management App uses the CWMP protocol on the CPE and ACS and is locally defined by device vendors.

RPC Methods define a generic mechanism by which an ACS can read or write parameters to configure a CPE device and monitor CPE status and statistics. Each parameter consists of a "name-value" pair where "name" identifies the particular parameter and has a hierarchical structure similar to files in a directory,

with each level separated by a dot. The "value" of a parameter is one of several defined data types. RPC Methods also define a mechanism to facilitate file downloads or optionally uploads for a variety of purposes, which includes firmware upgrades or vendor-specific configuration files. File transfers can be performed by Unicast (for downloads) or multicast transport protocols. Unicast protocols include HTTP/HTTPS, FTP, SFTP, and TFTP. Multicast protocols include FLUTE and DSM-CC.

RPC Methods are encoded using standard XML-based syntax called SOAP. CWMP recommends using the SOAP 1.1 protocol. SOAP runs over standard HTTP 1.1 protocol and security is enabled using SSL 3.0 and TLS 1.0 specifications, which run over standard TCP/IP protocol. The CPE device acts as the HTTP client and the ACS acts as the HTTP server. HTTP can be run directly over TCP/IP if security is not a major concern.

## Protocol Operation

The CPE device establishes a connection toward the ACS when it boots up for the first time or at a suitable trigger such as a change in the URL of the ACS. The ACS can be configured in the CPE or alternatively the CPE can discover the ACS using DHCP. After identifying the CPE (using the CPE vendor, model, software version, or other criteria), the ACS configures the writeable parameters in the CPE using RPCs. All of these configurations take the form of request/response and form a transaction, hence a series of transactions might be required to configure the CPE based on the number of parameters to be configured. All these transactions can happen in a single TCP/IP connection or they could span across multiple TCP/IP connections.

An event is an indication that something of interest has happened that requires the CPE to notify the ACS via an Inform request. The CPE must attempt to deliver an event at least once. If the CPE is not currently in a session with the ACS, it must attempt to deliver events immediately by initiating a session with the ACS, otherwise it must attempt to deliver them after the current session terminates.

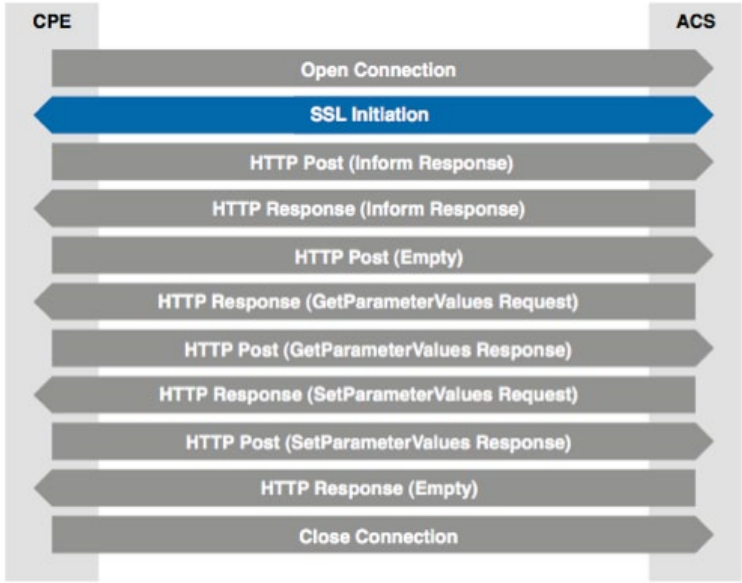


**Figure 6.** *Transaction Session Example*[1]

All transaction sessions must begin with an Inform message from the CPE to the ACS contained in the initial HTTP POST. This will be used to initiate a set of transactions and to communicate the limitations of the CPE with respect to message encoding. The session terminates when both the ACS and CPE have no more requests to send and no responses pending from either the ACS or the CPE. Only one transaction session shall exist between the CPE and ACS at a time. Please refer to Figure 6 for an example Transaction Session.

## FAP Data Model

The Femto Forum is driving definition of the FAP Data Model along with BBF's Working Group 3. Reference[9] outlines the current working draft (submitted as a contribution to BBF) of the FAP Data Model as defined by the Femto Forum. The FAP Data Model is being defined based on the TR-106 template and will be transported over the CWMP protocol defined in TR-069. Much of the following text is part of the current working draft.

Figure 7 illustrates the FAP Data Model, which is defined in Reference[9] as a Service object and called FAPService. FAPService is a container for a collection of three broad categories of management objects that cover all the aspects of FAP management. The base FAPService object includes parameters to identify whether the FAP service instance is enabled and the number of RF instances supported. The following sections briefly describe these management objects; refer to Reference[9] for details on the parameters.

## Control Object Group

This group covers all the objects and parameters required to control the FAP's operation including:

- *FapDevice* object contains the general product description and hardware capabilities including FAP product Type. Parameters in this section are read-only. It contains two objects:
  - *Description* object contains parameters characterizing the general product description including FAP Type (i.e., standalone or integrated), Vendor Name, Hardware Name and Version, Device ID, and other customer-specific information.
  - *CapabilitySet* object contains parameters characterizing the capabilities supported by the FAP such as whether it is equipped with GPS; its maximum transmit power; whether it supports GSM, HSDPA, HSUPA, FDD/GSM; etc.
- *FapControl* object contains state management of the FAP and associated control of the FAP done by the network side. The base FapControl object contains parameters characterizing the general FAP state such as whether the FAP is enabled, whether FAP administration is locked, whether the RF transmitter is enabled, and the type of System supported (e.g., Umts). Device state management is based on X.731. FapControl object in turn contains one object:
  - *Umts* object contains parameters that identify the Femto Gateway (FGW) and Security Gateway (SeGW) host name and IP address.

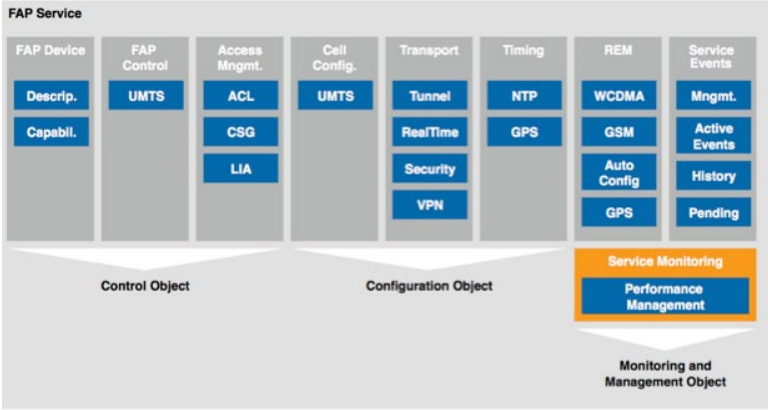


**Figure 7.** *FAP Data Model Structure*[9]

In the future, new objects for additional RF technology such as CDMA2000 and GSM are expected to be added to the FAPControl base object.

- *AccessManagement* object ensures management of subscription-based information. The base object contains parameters to identify whether Access Control List (ACL), Closed Subscriber Group (CSG), and Local IP Access (LIA) are supported. In addition it contains three objects:
  - *ACL* object parameters identify the maximum number of ACL entries supported and the International Mobile Subscriber Identity (IMSI) and Temporary Mobile Subscriber Identity (TMSI) numbers for each ACL entry.
  - *CSG* object parameters are yet to be defined and would likely identify a 'whitelist' of subscribers in a group.
  - *LIA* object parameters are yet to be defined and would likely identify local IP access domain and devices included among other parameters.

## Configuration Object Group

This group covers the aspects to configure the FAP for proper operation, including:

- *CellConfig* object contains configuration management of FAP functions and protocols. The base object contains parameters to identify the type of protocol supported (e.g., Umts). In addition, this object contains:
  - *Umts* object that contains parameters to enumerate CN/RAN/Cell (RF) related configurations that define the FAP operational parameters, e.g., Public Land Mobile Network (PLMN) type (GSM or ANSI), Mobile Country Code (MCC), Mobile Network Code (MNC), Radio Network Controller (RNC) identifier, Cell Mode (FDD or TDD), Cell identifier, etc.

In the future, new objects for additional RF technology such as CDMA2000 and GSM are expected to be added to the CellConfig base object.

- *Transport* object contains parameters and objects to manage functions and protocols related to Transport between the FAP and FGW. It contains four main objects
  - *Tunnel* object is related to IPsec tunnels. The IPsec Architecture (see RFC4301) describes a general processing model based on three nominal databases: (1) Security Policy Database (SPD) specifying the policy that applies to all IP traffic (inbound or outbound); (2) Security Association Database (SAD) that contains parameters associated with each Security Association (SA); and (3) Peer Authorization Database (PAD) that provides a link between an SA management protocol (such as IKE) and the SPD. SPD is modeled by the TR-098 [2] Queue Management object, and mainly by the Classification table. SAD and PAD are modeled by Tunnel object parameters such as IKE SA peer address, IKE SA creation time, IKE SA lifetime, Child SA creation time, Child SA life time, etc.
  - *RealTime* object manages Real Time Protocol (RTP) session and stream information via two tables. The RTP session table maintains an entry for each session with information such as session status, peer address, peer port, etc. The RTP stream table maintains an entry for sender and receiver with information such as stream status, RTCP status, stream direction, stream lost packet count, etc.
  - *Security* object contains parameters and objects to manage security key information. It contains two tables: (1) The Shared-secret table gathers information about all types of shared-secret based credentials (e.g., simple shared secrets, UICC, emulated UICC, etc.); (2) The Public Key table gathers information about all types of public key based elements (e.g., raw key pairs, X.509 certificates, etc.) and stores CPE credentials, trust-anchors, etc. Basic X.509 certificate management is also supported.
  - *VPN* object contains parameters relating to IPSec Tunnel based connectivity, including IP Address, subnet mask, DNS servers, count of bytes received/sent, etc., for the VPN.
- *Timing* object contains management of synchronization mode, such as NTP and GPS. This object is yet to be defined further.

## Monitoring and Management Object Group

This group covers the aspects to monitor the operation of the FAP:

- *REM* object contains measurement of the mobile network information in the surrounding environment. The base object contains parameters such as Radio Environment Measurement (REM) trigger event, frequency of REM trigger, etc. In addition, it contains four objects:
  - *WcdmaFdd* object represents the information gathered by the FAP by monitoring the RF environment in FDD mode of WCDMA in a Umts system. The collected information includes the Umts WCDMA FDD cells (both regular NodeB macrocells as well as other FAPs in the area). It contains parameter- enumerating information such as whether REM is enabled, timestamp of last REM, number of measured cells, etc.
  - *Gsm* object represents the information gathered by the FAP by monitoring the RF environment in a GSM system including the FAP capable of receiving the GSM band. Parameters for this object are very similar to the parameters in WcdmaFdd object.
  - *AutoConfig* object contains potential parameter values that are self-derived by the FAP based on the REM above.
  - *Gps* object contains GPS-derived location information (such as Latitude, Longitude, last measured time, etc.) when the FAP contains a GPS receiver.
- *ServiceEvents* object contains the FAP event management related information to be delivered to the ACS. It contains four objects:
  - *Management* object maintains the configuration of FAP service events such as event type, probable cause of the event, perceived severity of the event, authentication credential to upload specific event file, etc.
  - *ActiveEvents* object maintains a list of active events (i.e., having an associated lifecycle) on the FAP such as event type, timestamp when the event was raised, probable cause of the event, perceived severity, etc.
  - *History* object maintains a cyclic history of generated events on the FAP such as event type, timestamp when the event was raised, probable cause of the event, perceived severity, etc.
  - *PendingDelivery* object maintains the event messages waiting to be delivered to the upstream Element Management System such as event type, timestamp when the event was raised, probable cause of the event, perceived severity, etc.
- *ServiceMonitoring* object contains aggregation of periodic sampling of counters in the UTRAN control function or the WCDMA cell supported by a FAP for statistics purpose, such as the number of successfully established Radio Access Bearer (RAB) connections, number of RABs successfully modified or released by the FAP, etc.

## References

[1] DSL Forum TR-069 Amendment 2, "CPE WAN Management Protocol v1.1," Dec. 2007

[2] DSL Forum TR-098 Amendment, "Internet Gateway Device Data Model for TR-069," Nov. 2006

[3] DSL Forum TR-106 Amendment 1, "Data Model Template for TR-069-Enabled Devices," Nov. 2006

[4] 3GPP TS33.210, "Network Domain Security (NDS); Authentication Framework (AF) (Rel. 8)." Mar. 2008

[5] 3GPP TS32.642, "Configuration Management (CM); UTRAN network resources IRP (IRP); Network Resource Model (NRM) (Rel.8)," Mar. 2008

[6] 3GPP TS32.405, "Performance Management (PM); Performance measurements UTRAN (UTRAN) (Rel. 8)," Mar. 2008

[7] 3GPP TS22.011, "Service Accessibility (Rel. 8)," June 2008

[8] 3GPP TR25.820, "3G Home NodeB Study Item Technical Report (Rel. 8)," Mar. 2008

[9] ff_wg3_Data_Model_draft_v05, "FAP Data Model (Working Draft),", Femto Forum Sep. 2008

radisys.

**Corporate Headquarters**
5435 NE Dawson Creek Drive
Hillsboro, OR 97124 USA
503-615-1100 | Fax 503-615-1121
Toll-Free: 800-950-0044
www.radisys.com | info@radisys.com

intel Embedded Alliance Premier

©2011 Radisys Corporation.
Radisys, Trillium, Continuous Computing and Convedia are registered trademarks of Radisys Corporation.
*All other trademarks are the properties of their respective owners.
September 2011