RSA® Authentication Manager 8.0
RADIUS リファレンス ガイド

商標について

RSA、RSA のロゴ、EMC は、EMC Corporation の登録商標または商標です。その他のすべての名称ならびに製品についての商標は、それぞれの所有者の商標または登録商標です。RSA の商標の最新のリストについては、**http://japan.emc.com/legal/emc-corporation-trademarks.htm#rsa** を参照してください。

使用許諾契約

本ソフトウェアと関連ドキュメントは、EMC が著作権を保有しており、ライセンス契約に従って提供されます。本ソフトウェアと関連ドキュメントの使用と複製は、ライセンス契約の条項に従い、上記の著作権を侵害しない場合のみ許諾されます。本ソフトウェアと関連ドキュメント、およびその複製物を他人に提供することは一切認められません。

本ライセンス契約によって、本ソフトウェアと関連ドキュメントの所有権およびその他の知的財産権が譲渡されることはありません。本ソフトウェアと関連ドキュメントを不正に使用または複製した場合、民事および刑事責任が課せられることがあります。
本ソフトウェアは予告なく変更されることがありますので、あらかじめご承知おきください。

サード パーティ ライセンス

本製品は RSA 以外のサードパーティ製ソフトウェアを実装している場合があります。本製品内のサード パーティ製ソフトウェアに適用される使用許諾契約書の内容については、製品 DVD の Third-Party Licenses フォルダに含まれる情報を参照してください。

暗号技術に関するご注意

本製品には、暗号技術が組み込まれています。これらの暗号技術の使用、輸入、輸出は、各国の法律で禁止または制限されています。本製品を使用、輸入、輸出する場合は、各国における使用または輸出入に関する法律に従わなければなりません。

免責事項

本書に記載されている EMC ソフトウェアを使用、複製、および配布するには、適切なソフトウェア ライセンスが必要です。

EMC Corporation は、この資料に記載される情報が、発行日時点で正確であるとみなしています。この情報は予告なく変更されることがあります。

この資料に記載される情報は、「現状有姿」の条件で提供されています。EMC Corporation は、本文書に記載される情報に関する、どのような内容についても表明保証条項を設けず、特に、商品性や特定の目的に対する適応性に対する黙示の保証はいたしません。

2013 年 3 月

目次

はじめに

本書について

本書では、RSA RADIUS で使用する初期設定ファイル、Dictionary ファイル、構成ファイルの使用法および設定について説明しています。

本書は、管理者およびその他の信頼できる担当者のみを対象としています。一般ユーザー向けではないことにご注意ください。

RSA Authentication Manager 8.0 マニュアル

RSA Authentication Manager 8.0 の詳細については、次のマニュアルを参照してください。RSA では、管理者のみがアクセス可能なネットワーク上の場所に製品マニュアルを保管することを推奨します。

リリース ノート。本リリースにおける新機能や変更点に加え、既知の不具合とその回避方法について説明しています。

Getting Started：Authentication Manager の Quick Setup プロセスを実行する方法について説明しています。

設計ガイド：Authentication Manager の上位レベルのアーキテクチャ、および既存ネットワークとの統合方法について説明しています。

セットアップと構成ガイド： Authentication Manager をセットアップおよび構成する方法について説明しています。

管理者ガイド： Authentication Manager とその機能の概要を示しています。システムを構成し、ユーザーやセキュリティ ポリシーの管理など、さまざまな管理タスクを実行する方法について説明しています。

Help Desk Administrator's Guide：ヘルプ デスク管理者が日々実行する最も一般的なタスクの手順について説明しています。

SNMP Reference Guide：.SNMP（Simple Network Management Protocol）を構成して Authentication Manager のインスタンスを監視する方法について説明しています。

トラブルシューティング ガイド：RSA Authentication Manager の最も一般的なエラー メッセージと、各イベントをトラブルシューティングする適切なアクションについて説明します。

Developer's Guide：RSA Authentication Manager API（アプリケーション プログラミング インタフェース）を使用したカスタム プログラムの開発について説明しています。また、API の概要および Java API の Javadoc も含まれています。

Performance and Scalability Guide：最適なパフォーマンスを得るために環境を調整するときの考慮事項について説明しています。

***6.1* から *8.0* への移行ガイド**：RSA Authentication Manager 6.1 環境から RSA Authentication Manager 8.0 環境へ移行する方法について説明しています。

***7.1* から *8.0* への移行ガイド**：RSA Authentication Manager 7.1 環境から RSA Authentication Manager 8.0 環境へ移行する方法について説明しています。

Security Console ヘルプ：Security Console で日々実行される管理タスクについて説明しています。

Operations Console ヘルプ：Operations Console で実行される構成タスクおよび設定タスクについて説明しています。

Self-Service Console ヘルプ：Self-Service Console を使用する方法について説明しています。ヘルプを表示するには、Self-Service Console の［**Help（ヘルプ）**］タブで［**Self-Service Console Help（Self-Service Console ヘルプ）**］をクリックします。

RSA Token Management Snap-In ヘルプ：Active Directory アイデンティティ ソースが存在する環境で MMC（Microsoft 管理コンソール）と連携するソフトウェアを使用する方法について説明します。このスナップインを使用すると、トークンの有効化 / 無効化、トークンの割り当て、またはその他のトークン関連タスクを実行するために、Security Console にログオンする必要がなくなります。

関連ドキュメント

***RADIUS* リファレンス ガイド**： RSA RADIUS で使用する初期設定ファイル、Dictionary ファイル、構成ファイルの使用法および設定について説明しています。

Security Configuration Guide：Authentication Manager で使用可能なセキュリティ構成設定について説明します。また、安全な導入と使用のための設定、安全な保守、物理的なセキュリティ管理に関する情報も提供します。

サポートとサービス

RSA SecurCare Online	**https://knowledge.rsasecurity.com**
RSA Solution Gallery	**https://gallery.emc.com/community/marketplace/rsa?view=overview**

RSA SecurCare Online では、よくある質問に対する回答や既知の問題に対する解決策を含むナレッジベースを参照できます。また、新しいリリースに関する情報や重要なテクニカル ニュース、ソフトウェアのダウンロードも利用できます。

RSA Solution Gallery では、RSA 製品との連携が検証されているサードパーティのハードウェア製品およびソフトウェア製品に関する情報を利用できます。このギャラリーには、RSA 製品とこれらのサードパーティ製品の相互運用について、連携手順を説明した Secured by RSA 実装ガイドおよびその他の情報が含まれています。

カスタマー サポートにご連絡される前に

ご連絡いただく際には、次の情報をご用意ください。

- ❑ RSA Authentication Manager アプライアンスにアクセスできることを確認してください。
- ❑ ライセンスのシリアル番号。シリアル番号は、次の方法で確認できます。
  - Security Console にログオンして、[**License Status（ライセンス ステータス**]をクリックします。[**View Installed License（インストール済みライセンスの表示)**]をクリックし、リストからライセンス ID を選択して内容を表示します。
- ❑ Authentication Manager アプライアンスのソフトウェアのバージョン情報。この情報は、Quick Setup の画面の右上隅、または Security Console に表示されます。Security Console にログオンして、[**Software Version Information（ソフトウェア バージョン情報)**]をクリックします。

1 radius.ini ファイル

radius.ini 初期設定ファイルは、RSA RADIUS の動作を決定する主要な構成ファイルです。この構成ファイルには、RADIUS のさまざまな機能と動作を制御する情報が格納されます。

注：構成ファイルには、現在のリリースの RSA RADIUS では使用しないパラメータやセクションが含まれていることがあります。本書では、現在のリリースに関連しているセクションおよびパラメータについてのみ説明しています。本書で説明していないパラメータは変更しないでください。

注：radius.ini ファイル内の設定を編集する場合は、標準的な .ini 構文に準拠する必要があります。**radius.ini** ファイルに何らかの変更を加えた後は、それらの変更を反映するために RADIUS サーバを再起動する必要があります。

［Addresses］セクション

RSA RADIUS サーバのデフォルト動作では、RSA RADIUS サーバがすべての使用可能なネットワーク インタフェース上で着信 RADIUS パケットをリスンできるよう、RSA RADIUS が稼働しているサーバのプライマリ ホスト名に対してネーム サービスから報告されるすべての IPv4 アドレスを自動構成します。

デフォルト設定の AutoConfigureIPv4 を指定した場合、RSA RADIUS は、ローカル ホストのすべての IPv4 アドレスを自動的に検出して構成しようとします。

［AuditLog］セクション

［AuditLog］セクションでは、管理者アクティビティおよび CCM イベントを記録する監査ログ ファイル（*yyyymmdd*.auditlog）を RSA RADIUS で維持するかどうかを指定します。監査ログ レコードは、xml 形式で保存されます。

管理者アクティビティには、次のアクティビティが含まれます。

- RSA RADIUS 管理者によるログオンとログオフ
- RSA RADIUS オブジェクト（RADIUS クライアント、ユーザー、プロファイル、CCM ノード）の作成、変更、削除
- ファイルのインポート

CCM イベントには、CCM ファイルの発行、通知、ダウンロードが含まれます。

```
[AuditLog]
;Enable = 0
;LogfilePermissions = owner:group mode
;DaysToKeep = 30
```

次の表は、［AuditLog］のパラメータとそれらの機能を示しています。

パラメータ	機能
Enable	• 0 に設定すると、監査ログが無効になる。 • 1 に設定すると、監査ログが有効になる。 デフォルト値は 0。
LogfilePermissions	監査ログ（*yyyymmdd*.auditlog）ファイルのオーナーとアクセス パーミッションを指定する。 LogFilePermissions の値は、オーナー : グループ パーミッションの形式で入力する。 • オーナー：ファイルの所有者をテキスト形式または数値形式で指定する。 • グループ：ファイルのグループ設定をテキスト形式または数値形式で指定する。 • パーミッション : オーナー / グループ / その他のユーザーがファイルに対してどの権限を行使できるかをテキスト形式または数値形式で指定する。 たとえば、ralphw:1007 rw-r- - - - と指定した場合は、ファイル所有者（ralphw）に監査ログ ファイルの読み取りと編集の権限が与えられ、グループ 1007 のメンバーに監査ログ ファイルの読み取りの権限が与えられる（編集の権限はなし）。ただし、その他のユーザーは監査ログ ファイルにアクセスできない。
DaysToKeep	各認証受け付けレポートを RSA RADIUS サーバで保持する日数を指定する。 デフォルト値は 30 日。

［Configuration］セクション

radius.ini の［Configuration］セクションには、RSA RADIUS の基本動作を制御するパラメータが含まれています。

次の表は、［Configuration］のパラメータとそれらの機能を示しています。

パラメータ	機能
AddDestIPAddressAttrToRequest	• 0 に設定した場合、RSA RADIUS は宛先アドレス情報を RADIUS 要求に追加しない。 • 1 に設定した場合、RSA RADIUS は RADIUS 要求の宛先となった IP アドレスを識別する Funk-Dest-IP-Address 属性をパケットに追加する。要求パケットに含まれている属性に対して実行可能なすべての処理（チェックリストの処理など）を、この属性に対して実行することができる。 デフォルト値は 0。
AddDestUDPPortAttrToRequest	• 0 に設定した場合、RSA RADIUS は宛先ポート情報を RADIUS 要求に追加しない。 • 1 に設定した場合、RSA RADIUS は RADIUS 要求の宛先となった UDP ポートを識別する Funk-Dest-UDP-Port 属性をパケットに追加する。要求パケットに含まれている属性に対して実行可能なすべての処理（チェックリストの処理など）を、この属性に対して実行することができる。 デフォルト値は 0。
AddSourceIPAddressAttrToRequest	• 0 に設定した場合、RSA RADIUS はソース アドレス情報を RADIUS 要求に追加しない。 • 1 に設定した場合、RSA RADIUS はどの IP アドレスから RADIUS 要求を受信したかを識別する Funk-Source-IP-Address 属性をパケットに追加する。要求パケットに含まれている属性に対して実行可能なすべての処理（チェックリストの処理など）を、この属性に対して実行することができる。 デフォルト値は 0。

パラメータ	機能
Apply-Login-Limits	• yes に設定した場合、ユーザーごとの同時接続の最大数が制限され、制限を超過する接続試行は拒否される。 • no に設定した場合は、制限を超える接続が許可されるが、サーバ ログ ファイルにイベントが記録される。 デフォルト値は yes。
AuthenticateOnly	• 1 に設定した場合は、AuthenticateOnly（Service-Type 8）要求に対する応答パケットに応答属性が含められない。 • 0 に設定した場合は、通常の応答属性が応答に含められる。 デフォルト値は 1。
CheckMessageAuthenticator	ネットワーク アクセス デバイスから Access-Request を受信した際、またはプロキシ（拡張プロキシのみ）から Access-Accept、Access-Reject、Access-Challenge のいずれかを受信したときに Message-Authenticator の検証を行うかどうかを指定する。 • 0 に設定した場合は、受信した Message-Authenticator 属性の検証を行わない。 • 1 に設定した場合は、受信した Message-Authenticator 属性の検証を行う。 デフォルト値は 0。 **注：**通常のプロキシの場合、検証は行われません。

パラメータ	機能
ClassAttributeStyle	• 1 に設定した場合、RSA RADIUS は Access-Accept パケット内で、複数の ASCII キーを持つ非暗号化形式の Class 属性を使用する。 • 2 に設定した場合、RSA RADIUS は Access-Accept パケット内で拡張 / 暗号化形式の Class 属性を使用する。 デフォルト値は 2。 **注：**属性の埋め込みを行う前に、ClassAttributeStyle パラメータの値を 2 に設定しておく必要があります。属性の埋め込みについては、20 ページの「[Debug] セクション」を参照してください。
DisableSecondaryMakeModelSelection	• 1 に設定した場合、RSA RADIUS は要求のソース アドレスを使用してネットワーク アクセス デバイス エントリーを検索し、クライアントに対して指定されている情報に従って製造元 / モデルを設定する。 • 0 に設定した場合、RSA RADIUS は次のように動作する。 1. 要求のソース アドレスを使用してネットワーク アクセス デバイス エントリーを検索し、クライアントに対して指定されている情報に従って製造元 / モデルを設定する。 2. NAS-IP-Address 属性（存在する場合）を使用して、ネットワーク アクセス デバイス エントリーを検索する。IP アドレスが見つかった場合は、手順 1 で特定した製造元 / モデル情報を上書きする。 3. NAS-Identifier 属性（存在する場合）を使用して、ネットワーク アクセス デバイス エントリーを名前で検索する。名前が見つかった場合は、手順 1 または手順 2 で特定した製造元 / モデル情報を上書きする。 デフォルト値は 0。

パラメータ	機能
EnhancedDiagnosticLogging	• no に設定すると、標準的な診断ログ メッセージが RADIUS ログ ファイルに書き込まれる（ログ レベルが 0 に設定されている場合）。 • yes に設定すると、標準的な診断ログ メッセージに加えて、プロキシ再試行、プロキシ タイムアウト、LDAP タイムアウトに関するメッセージが RADIUS ログ ファイル（*yyyymmdd*.log）に書き込まれる（ログ レベルが 0 に設定されている場合）。 デフォルト値は no。
FramedIPAddressHint	yes に設定すると、Framed-IP-Address 属性がヒントとして扱われる。ユーザーのリターンリストがプールから Framed-IP-Address を割り当てるように構成されていても、この属性が Access-Request に含まれている場合は、IP アドレスを新たに割り当てるのではなく Access-Request の IP アドレスが返される。 デフォルト値は no。
LogAccept	• 1 に設定した場合は、Accepts に関連するメッセージのうち、現在の LogLevel に一致するメッセージがサーバ ログ ファイルに記録される。 • 0 に設定した場合、Accepts に関連するメッセージは無視される。 デフォルト値は 1。

パラメータ	機能
LogFilePermissions	システム ログ（*yyyymmdd*.log）ファイルのオーナーとアクセス パーミッションを指定する。 LogFilePermissions の値は、オーナー : グループ パーミッションの形式で入力する。 • オーナー：ファイルの所有者をテキスト形式または数値形式で指定する。 • グループ：ファイルのグループ設定をテキスト形式または数値形式で指定する。 • パーミッション : オーナー / グループ / その他のユーザーがファイルに対してどの権限を行使できるかをテキスト形式または数値形式で指定する。 たとえば、ralphw:1007 rw-r- - - - と指定した場合は、ファイル所有者（ralphw）にログ ファイルの読み取りと編集の権限が与えられ、グループ 1007 のメンバーにログ ファイルの読み取りの権限が与えられる（編集の権限はなし）。ただし、その他のユーザーはログ ファイルにアクセスできない。
LogfileMaxMBytes	• 0 に設定した場合（または設定を省略した場合）、サーバ ログ ファイルのサイズは無視され、ログ ファイルを開いた日付がログ ファイル名に使用される（*YYYYMMDD*.log）。 • 1 ～ 2047 の範囲内の値に設定すると、現在のサーバ ログ ファイルが指定サイズ（単位：メガバイト（1024×1024 バイト））に達した時点で現在のログ ファイルが閉じられ、新しいログ ファイルが開かれる。新しいログ ファイルのファイル名には、そのファイルを開いた日時が使用される（*YYYYMMDD_HHMM*.log）。 デフォルト値は 0。 **注：**ログ ファイルのサイズは、1 分おきにチェックされます。ログ ファイルは、次のログ サイズ チェックが行われるまでロールオーバーされません。このため、ログ ファイルのサイズは、LogFileMaxMBytes の指定値を超えることがあります。

パラメータ	機能
LogHighResolutionTime	• no に設定した場合は、RSA RADIUS ログ ファイル（*yyyymmdd*.log）内のエントリーのタイムスタンプが *hh:mm:ss*（時 : 分 : 秒）の形式で記録される。 • yes に設定した場合は、RSA RADIUS ログ ファイル（*yyyymmdd*.log）内のエントリーのタイムスタンプが *hh:mm:ss:xxx* の形式で記録される。ここで *xxx* は、*ss* 値の変更後に経過した時間をミリ秒単位で表す。 デフォルト値は no。
LogLevel	RSA RADIUS がサーバ ログ ファイル（.LOG）にエントリーを書き込むログ レベルを設定する。 • 0：本番ログ レベル • 1：情報ログ レベル • 2：デバッグ ログ レベル デフォルト値は 0。
LogReject	• 0 に設定した場合、Rejects に関連するメッセージは無視される。 • 1 に設定した場合は、Rejects に関連するメッセージのうち、現在の LogLevel に一致するメッセージがサーバ ログ ファイルに記録される。 デフォルト値は 1。
NoNullTermination	• 0 に設定した場合、string 型の RADIUS 応答属性は、文字列の末尾にヌル文字を付加して（ヌル終端文字列にして）送信される。 • 1 に設定した場合、string 型の RADIUS 応答属性は、文字列の末尾にヌル文字を付加せずに送信される。このパラメータの値として 1 を入力すると、すべての応答属性を string 型から stringnz 型に変更した場合と同じ結果が得られる。 デフォルト値は 0。 **注：**この設定を変更した場合は、**saved-dicts.bin** ファイルを削除したうえで、RSA RADIUS サービスを再起動する必要があります。

パラメータ	機能
PhantomTimeout	ファントム セッション レコードをアクティブにする最長時間（秒数）。ファントム レコードは、対応するアカウンティング開始パケットが受信されると、ただちに破棄される。ファントム レコードがタイムアウト時間の経過後にまだ存在している場合は、ファントム レコードが破棄され、それに関連づけられているすべてのリソースが解放される。
PrivateDir	データベースおよび Dictionary ファイルが格納されている RSA RADIUS ディレクトリの場所の名前。 RSA RADIUS サービス / デーモンの存在するディレクトリがデフォルト値となる。
ProxySource	すべての発信プロキシ トラフィックを通過させるインタフェースの IP アドレスを指定する。ProxySource で指定する IP アドレスは、**radius.ini** の［Addresses］セクションに含まれている必要がある。 ProxySource アドレスが指定されておらず、プロキシ インタフェースのレルム別制御が有効にされていない場合、RSA RADIUS はサーバ上で最初に検出したインタフェースを使用する。
ProxyStripRealm	• 1 に設定した場合は、要求を下流に送信する前にプロキシ レルム修飾が削除される。 • 0 に設定した場合、レルム名の削除は行われない。 デフォルト値は 1。

パラメータ	機能
SendOnlyOneClassAttribute	認証中にユーザーの識別情報を暗号化する際、RSA RADIUS は暗号化されたユーザーの識別情報をアカウンティング サーバに渡すために特殊な Class 属性を使用する。通常は、このような Class 属性を Accept 応答に複数含める必要があるが、アクセス ポイントによっては複数の Class 属性のエコー処理をサポートしていない。その場合は、SendOnlyOneClassAttribute パラメータを使用して、暗号化されたユーザー識別情報を RSA RADIUS でどのように転送するかを指定することができる。 • 1 に設定した場合、RSA RADIUS は Class 属性フラグ、サーバ識別子、トランザクション識別子を格納する Class 属性を作成する。この場合、通常は Class 属性に格納されるユーザー識別データが現在のセッション テーブルに格納される。RSA RADIUS はアカウンティング要求を受信した時点で、現在のセッション テーブルから Class 情報を参照し、その情報をアカウンティング要求パケットに含まれていた情報と同じように使用する。 • 0 に設定した場合、RSA RADIUS は暗号化されたユーザーの識別情報をアクセス ポイントに返すために 1 つ以上の Class 属性を作成する。このとき、アクセス ポイントが暗号化されたユーザー識別情報が格納されている Class 属性をアカウンティング サーバに転送できることが前提となる。 デフォルト値は 0。 **注：**この機能を利用できるのは、認証を実行するのと同じサーバにアカウンティング要求が渡される場合のみです。認証を実行するサーバとは異なるサーバにアカウンティング要求が渡される場合、この機能は使用できません。
StartupTimeout	起動シーケンスが完了するまで RSA RADIUS が待機する時間を秒単位で指定する。指定した時間に達すると、タイムアウトになる。 デフォルト値は 360 秒。

パラメータ	機能
TraceLevel	RADIUS パケットのトレース レベルを指定する。 • 0：パケット トレースなし • 1：パケットをパースしてログに記録 • 2：パケットの生データとパースした結果の両方をログに記録 デフォルト値は 0。 **注：**パケット トレースは、サーバ ログ ファイルに書き込まれ、相互運用性に関する問題のトラブルシューティングに役立ちます。
TreatAddressPoolsAsDisjoint	• 1 に設定した場合、RSA RADIUS は各 IP アドレス プールを互いに重複しないアドレス空間であるかのように扱う。この場合、IP アドレスが 1 つのアドレス プールにのみ割り当てられていることを検証する通常のチェックは無効になる。 • 0 に設定した場合は、1 つの IP アドレスを 1 つの IP アドレス プールから 1 つのセッションに割り当てることのみ可能となる。 デフォルト値は 0。 **注：**RSA RADIUS では、割り当て済みのリソースをトラッキングするために Class 属性を使用して IP アドレスをトラッキングします。この属性には、IP プール名と IP アドレスが格納されます。
UseNewAttributeMerge	• 1 に設定した場合は、新しいプロファイルとユーザー属性のマージ計算が実行される。 • 0 に設定した場合、古い計算法が使用される。 デフォルト値は 1。

［CurrentSessions］セクション

radius.ini の［CurrentSessions］セクションでは、現在のセッション テーブルを制御します。

```
[CurrentSessions]
;CaseSensitiveUsernameCompare = 1
```

次の表は、［CurrentSessions］のパラメータとそれらの機能を示しています。

パラメータ	機能
CaseSensitiveUsernameCompare	• 1 に設定した場合は、サーバが現在のセッション テーブルから同じユーザー名を持つセッションを検索する際、大文字と小文字を区別する。 • 0 に設定した場合、サーバは大文字と小文字を区別しない。 デフォルト値は 1。

［Debug］セクション

radius.ini の［Debug］セクションでは、RSA RADIUS の操作に関する問題のデバッグに役立つスレッド識別子をログ メッセージに組み込むことができます。異なる RADIUS 要求に関するメッセージが交互に記録されている場合は、スレッド識別子を使用すると診断ログを解析しやすくなります。

スレッド識別子を診断ログ メッセージに含めるには、次の構文を使用します。

```
[Debug]
Log-Thread-ID = yes
```

次の表は、［Debug］のパラメータとそれらの機能を示しています。

パラメータ	機能
Log-Thread-ID	• yes に設定すると、RSA RADIUS ログ メッセージにスレッド識別子が含まれる。 • no に設定すると、RSA RADIUS ログ メッセージからスレッド識別子が省略される。 デフォルト値は no。

［EmbedInClass］セクション

radius.ini の［EmbedInClass］セクションでは、認証処理中に使用可能な属性のうち、アカウンティング要求内で使用できるようにする必要のある属性を指定します。属性の埋め込みにより、ネットワーク アクセス デバイスから RSA RADIUS へ返される Class 属性の中に課金情報を埋め込むことができます。RSA RADIUS は埋め込み属性を受信すると、その属性をデコードしてアカウンティング要求に含めます。このときの動作は、**classmap.ini** ファイル（49 ページの classmap.ini ファイルを参照）で指定された設定に従います。

注：属性の埋め込みを使用する前に、**radius.ini** の［Configuration］セクションで ClassAttributeStyle パラメータの値を 2 に設定しておく必要があります。

属性を埋め込むには、次の構文を使用します。

```
[EmbedInClass]
responseAttribute={ Clear | Encrypt }[,Remove]
```

次の表は、［EmbedInClass］のパラメータとそれらの機能を示しています。

パラメータ	機能
responseAttribute	RADIUS Class 属性に埋め込む応答属性を指定する。
Clear	取得した情報をクリア テキスト形式で Class 属性に含めるように指定する。
Encrypt	取得した情報を暗号化したうえで Class 属性に含めるように指定する。
Remove	省略可能なパラメータ。Accept-Response パケットから埋め込み属性を削除する。

［HiddenEAPIdentity］セクション

radius.ini の［HiddenEAPIdentity］セクションでは、認証要求への応答として返される Access-Accept メッセージに EAP/TTLS プロトコルおよび EAP/SIM プロトコルの既知の内部 ID を含めることができます。

構文は次のとおりです。

```
[HiddenEAPIdentity]
IncludeInAcceptResponse=0|1
ResponseAttribute = attributeName[, replaceAttribute]
```

次の表は、［HiddenEAPIdentity］のパラメータとそれらの機能を示しています。

パラメータ	機能
IncludeInAcceptResponse	• 0 に設定した場合は、Access-Accept 応答に内部 ID が含まれない。 • 1 に設定した場合は、Access-Accept 応答の属性のうち、指定した属性に内部 ID が含められる。 デフォルト値は 0。
attributeName	Access-Accept メッセージの属性のうち、内部 ID を含める属性を指定する。この値を省略すると、User-Name 属性が使用される。attributeName には、属性ディクショナリに指定されている任意の文字列属性（VSA など）を使用することができる。
[, replaceAttribute]	attributeName パラメータで指定した属性の元の値を保持する Access-Accept 属性を指定する。 置換する属性を指定しない場合、元の属性値は失われる。

［MsChapNameStripping］セクション

radius.ini の［MsChapNameStripping］セクションでは、エンド ユーザーから転送されたユーザー名 / パスワード ハッシュと RSA RADIUS のユーザー エントリーを照合するときに、ユーザー名からドメイン情報を削除するかどうかを指定します。エンド ユーザーのホストは、ユーザーの認証情報のハッシュを計算する前に、ユーザー名からドメイン情報を削除します。この機能は、RSA RADIUS データベースに格納されたユーザー名に、ドメイン情報とみなされる特定の文字が含まれている場合に役立ちます。

この機能を有効にした場合：

1. RSA RADIUS は、データベース内のユーザー名をスキャンし、ユーザー名の先頭にドメイン名が付加されている可能性を示すプレフィックス区切り文字を検索します。プレフィックス区切り文字が検出されると、サーバはその文字（およびその区切り文字の左側にあるすべての文字）を削除した後、ユーザーの認証情報のハッシュを独自に生成し、その結果をエンド ユーザーから転送されたハッシュ済み認証情報と比較して、一致するかどうかを確認します。
2. プレフィックス区切り文字が見つからなかった場合（またはプレフィックスを削除した後、ハッシュ済みの認証情報と一致しなかった場合）、RSA RADIUS はユーザー名をスキャンして、ユーザー名の末尾にドメインが付加されている可能性を示す区切り文字を検索します。サフィックス区切り文字が検出されると、サーバはその文字（およびその区切り文字の右側にあるすべての文字）を削除した後、ユーザーの認証情報のハッシュを独自に生成し、その結果をエンド ユーザーから転送されたハッシュ済み認証情報と比較して、一致するかどうかを確認します。
3. プリフィックス区切り文字とサフィックス区切り文字のいずれも見つからなかった場合（または、区切り文字が見つかっても、ハッシュ済みの認証情報と一致しなかった場合）、サーバはユーザー名文字列全体を使用して、認証情報のハッシュを生成し、その結果をエンド ユーザーから転送されたハッシュ済み認証情報と比較して、一致するかどうかを確認します。

［MsChapNameStripping］セクションの構文は次のとおりです。

```
[MsChapNameStripping]
Enable=1
Prefix=\\
Suffix=/@
```

次の表は、［MsChapNameStripping］のパラメータとそれらの機能を示しています。

パラメータ	機能
Enable	• 0 に設定するか省略した場合、この機能は無効になる。 • 1 に設定すると、この機能が有効になる。 デフォルト値は 0。
Prefix	プリフィックスから削除する 5 文字までの ASCII 文字のリスト。リストにスペースが含まれる場合は、リスト全体を引用符で囲む必要がある。 バックスラッシュまたは円記号を削除する場合は、二重に記入する（\\）。二重のバックスラッシュまたは円記号は、リスト内で 1 文字として扱われる。 デフォルト値は \\。
Suffix	サフィックスから削除する 5 文字までの ASCII 文字のリスト。リストにスペースが含まれる場合は、リスト全体を引用符で囲む必要がある。 バックスラッシュまたは円記号を削除する場合は、二重に記入する（\\）。二重のバックスラッシュまたは円記号は、リスト内で 1 文字として扱われる。 デフォルト値は /@。

［Ports］セクション

radius.ini の［Ports］セクションでは、RSA RADIUS で使用する UDP ポートを設定できます。

- **radius.ini** の［Ports］セクションで 1 つ以上の UDPAuthPort 設定を指定すると、サーバが認証要求をリスンするポートは、このセクションで指定したポート番号のみに限定されます。同様に、1 つ以上の UDPAcctPort 設定を指定すると、サーバがアカウンティング要求をリスンするポートは、それらのポートのみに限定されます。
  最大 4,096 個のポートを指定できます。この制限を超過すると、RADIUS 認証サブコンポーネントを初期化できなくなります。
- ［Ports］セクションで UDPAuthPort 設定が指定されておらず、**/etc/services** ファイルで radius サービスと radacct が指定されていない場合、サーバは UDP ポート 1645 および 1812 を認証要求のリスンに使用し、UDP ポート 1646 および 1813 をアカウンティング要求のリスンに使用します。

注：選択した UDP ポートのいずれかへのバインドが失敗すると、対応するサブコンポーネント（認証またはアカウンティング）が初期化できなくなります。

次の表は、［Ports］のパラメータとそれらの機能を示しています。

パラメータ	機能
TCPControlPort	SNMP 通信および CCM/ レプリケーション通信に使用する TCP ポートを指定する。 デフォルト値は 1812。
UDPAuthPort	認証に使用する UDP ポートを 1 つ以上指定する。複数のポートを使用する場合は、ポート番号を 1 行に 1 つずつ指定する。 デフォルト値は 1645 および 1812。
UDPAcctPort	アカウンティングに使用する UDP ポートを 1 つ以上指定する。複数のポートを使用する場合は、ポート番号を 1 行に 1 つずつ指定する。 デフォルト値は 1646 および 1813。
UDPProxyPortBlockLength	プロキシ RADIUS 通信に使用するポート番号範囲に含めるアドレスの数を指定する。 デフォルト値は 64。
UDPProxyPortBlockStart	プロキシ RADIUS 通信に使用するポート番号範囲の開始ポート番号を指定する。 デフォルト値は 28000。 **注：**デフォルト値を変更する場合は、ネットワーク上の well-known UDP ポートやその他の使用中の UDP ポートと重複しない番号範囲を選択してください。 **注：**ネットワーク ファイアウォールは、指定した番号範囲のポートが外部と通信できるように構成しておく必要があります。

次に例を挙げます。

```
[Ports]
SecureTcpAdminPort = 1813
SecureTcpAdminAddress = 192.168.12.15
TcpControlPort = 1812
TCPControlAddress = 192.168.15.55
UDPAuthPort = 1645
UDPAuthPort = 1812
UDPAcctPort = 1646
UDPAcctPort = 1813
UDPProxyPortBlockStart = 28000
UDPProxyPortBlockLength = 64
```

［SecurID］セクション

radius.ini の［SecurID］セクションには、ISDN ユーザー向けの RSA SecurID 認証に固有の項目が含まれます。このセクションの情報は、RSA RADIUS が SecurID 認証の成功後にユーザーの認証情報を一時的にキャッシュできるようにするために使用されます。このキャッシュ処理は、ユーザーのセッション中に 2 番目の ISDN B チャネルの認証を可能にするために必要となります。RSA RADIUS では、キャッシュされたトークンを使用して 2 番目のチャネルを認証します。

注：この機能を有効にしていない場合は、両方の B チャネルを結合した ISDN 接続を通じてユーザーが SecurID 認証を受けようとする際、SecurID のセキュリティ違反が発生して認証が失敗することになります。1 つの B チャネルのみを使用している ISDN ユーザーは影響を受けません。

次の表は、［SecurID］のパラメータとそれらの機能を示しています。

パラメータ	機能
CachePasscodes	• yes に設定すると、RSA SecurID パスコード キャッシュが有効になる。 • no に設定すると、RSA SecurID パスコード キャッシュが無効になる。 デフォルト値は no。
SecondsToCachePasscodes	キャッシュした SecurID パスコード（PIN およびトークン コード）を保持する秒数。 デフォルト値は 60 秒。

［UserNameTransform］セクション

［UserNameTransform］セクションでは、RADIUS 要求に含まれるユーザー名を受信時の形式から処理可能な形式に変換するためのルールを指定できます。ユーザーがネットワーク アクセス デバイスに渡すユーザー名の形式は、RADIUS サーバがプロキシ転送に適用する形式や Authentication Manager が要求する形式と異なっていることがあります。その場合は、この機能が役立ちます。

ユーザー名は、変換ルールに指定された入力形式と出力形式に基づき、入力文字列から出力文字列に変換されます。ユーザー名変換ルールは、RADIUS 要求に含まれているユーザー名に適用されます。RADIUS 要求内のユーザー名は、入力形式に基づいて解析されます。

- ユーザー名が入力形式に一致していない場合は、ルールが適用されず、ユーザー名は変更されません。

- ルールが適用される場合は、解析されたユーザー名の各要素が出力形式に基づいて変換され、変換されたユーザー名が構築されます。
  - Access-Accept（または Acct-Start/Acct-Stop）に含まれている User-Name が入力形式ルールと比較される。
  - User-Name がルールに一致していれば、出力形式に変換され、認証が続行される。
  - User-Name が入力形式に一致していない場合は、変換が行われず、認証が続行される。

RADIUS 側の処理では、元のユーザー名の代わりに変換後のユーザー名が使用されます。変換後のユーザー名は、要求に最初から含まれていたものとして扱われます。パケットをプロキシ転送するかどうかは、変換後のユーザー名に基づいて決定されます。さらに、すべての認証が変換後のユーザー名に基づいて行われます。

形式文字列として、任意の文字シーケンスを使用できます。変数を山かっこ（<>）で囲んで、形式文字列に含めることもできます。テキスト内に含まれるバックスラッシュまたは円記号（\）は、リテラル文字を表すエスケープ文字として扱われます。変数名にバックスラッシュまたは円記号を含めた場合は、エスケープ文字ではなく、通常の文字として扱われます。したがって、変数名に右山かっこ（>）を使用することはできません。

リテラル テキストには、変数の要素に使用されていない文字を使用する必要があります。英数字ではなく、スラッシュ（/）や @ 記号などの区切り文字を使用してください。

ユーザー名変換ルールは、認証パケットとアカウンティング パケットの一方または両方に適用できます。

```
[UserNameTransform]
In=<input format>
Out=<output format>
Authentication=< yes | no >
Accounting=< yes | no >
```

次の表は、［UserNameTransform］のパラメータとそれらの機能を示しています。

パラメータ	機能
In	ユーザー名の入力形式を指定する形式文字列。 <user>@<realm> など。
Out	ユーザー名の出力形式を指定する形式文字列。 <user> など。
Authentication	yes に設定すると、認証要求に対して変換が有効になる。 デフォルト値は Yes。

パラメータ	機能
Accounting	yes に設定すると、アカウンティング要求に対して変換が有効になる。 デフォルト値は Yes。
Proxy	yes に設定すると、プロキシ要求に対して変換が有効になる。 デフォルト値は Yes。

たとえば、次の設定を使用すると、george@acme.com は george に変換されます。

```
In = <user>@<realm>
Out = <user>
```

次の設定を使用すると、abc/martha@bigco.com は bigco.com::abc/martha に変換されます。

```
In = <prefix>/<user>@<realm>
Out = <realm>::<prefix>/<user>
```

2 sbrd.conf ファイル

sbrd.conf ファイルは、実行可能な Bourne シェル スクリプトです。このスクリプトは、RSA RADIUS の実行環境を初期化するときに、**sbrd** プロセスによって起動されます。

注：構成ファイルには、現在のリリースの RSA RADIUS では使用しないパラメータやセクションが含まれていることがあります。本書では、現在のリリースに関連しているセクションおよびパラメータについてのみ説明しています。本書で説明していないパラメータは変更しないでください。

注：sbrd スクリプトは変更しないでください。ファイル マスク、ulimit、watchdog プロセスを変更するなど、適切な動作構成を指定するには、sbrd スクリプトの代わりに **sbrd.conf** を変更してください。

次に例を挙げます。

```
#!/bin/sh
###############
# sbrd.conf
###############
ULIMIT_CORE_SIZE=""
ULIMIT_CORE_COUNT=3
ULIMIT_OPEN_FILES=1024

RADIUSUMASK=""
RADIUS_HIGH_FDS=1
ORACLE_MSB_FILE=”ORACLE_HOME/rdbms/mesg/ocius.msb”

# RADIUS の実行可能プログラム、オプション、引数
RADIUS="radius"
RADIUSOPTS=""
RADIUSARGS="sbr.xml"
RADIUS_PRIVATE_DIR="$RADIUSDIR"

#watchdog プロセスの実行可能プログラム、オプション、引数
WATCHDOGENABLE=0
WATCHDOG="radiusd"
WATCHDOGOPTS="--config $RADIUSDIR/radiusd.conf --pidfile
$RADIUSDIR/radius.pid"
WATCHDOGARGS="$RADIUSDIR/$SELF"
```

注：sbrd.conf ファイル内のパラメータ設定には、スペースを含めないでください。

正しい例：ULIMIT_CORE_COUNT=3
誤った例：ULIMIT_CORE_COUNT = 3

次の表は、**sbrd.conf** のパラメータとそれらの機能を示しています。

パラメータ	機能
ULIMIT_CORE_SIZE	RSA RADIUS の実行が失敗したときに生成されるコア ファイルのサイズを指定する。 • ULIMIT_CORE_SIZE に値を指定した場合、その値は 1024 バイト ブロックを単位とするコア ファイルの最大サイズとなる。 • この値を disabled に設定した場合、RSA RADIUS は現在の環境を変更しないで使用する。 • この値を ““（スペースを含まない 2 つの二重引用符）に設定した場合、RSA RADIUS は現在の環境を使用するが、必要に応じて調整が行われる。 デフォルト値は ““。
ULIMIT_CORE_COUNT	RSA RADIUS サーバ上に保持するコア ファイルの数を指定する。サーバ上に存在するコア ファイルがすでに最大数に達しているときに RSA RADIUS の実行が失敗すると、最も古いコア ファイルが破棄され、新しいコア ファイルが作成される。 • この値を 0 ～ 999,999,999 の範囲内に設定した場合は、指定した数のコア ファイルがサーバ上に保持される。 • この値を unlimited に設定すると、RSA RADIUS は古いコア ファイルを破棄せずに新しいコア ファイルを生成する。 • この値を disabled に設定した場合、RSA RADIUS は現在の環境を変更しないで使用する。 • この値を ““（スペースを含まない 2 つの二重引用符）に設定した場合、RSA RADIUS は現在の環境を使用するが、必要に応じて調整が行われる。 デフォルト値は 3。
ULIMIT_OPEN_FILES	RSA RADIUS プロセスで同時に開くことができるファイルの数を指定する。 • この値を 256 ～ 1024 の範囲内に設定した場合は、指定した数のファイルが開いた状態でサーバ上に保持される。 • この値を disabled に設定した場合、RSA RADIUS は現在の環境を変更しないで使用する。 • この値を ““（スペースを含まない 2 つの二重引用符）に設定した場合、RSA RADIUS は現在の環境を使用するが、必要に応じて調整が行われる。 デフォルト値は 1024。

パラメータ	機能
RADIUSMASK	新しいログ ファイルの作成時に適用するファイル パーミッションを指定する。 • この値に umask 引数を指定すると、オーナー、グループ、その他のユーザに対して指定したパーミッションで、ログ ファイルが作成される。 • この値に “” を設定すると、その環境の umask によってオーナー、グループ、その他のユーザーに対してデフォルトのパーミッションを適用してログ ファイルが作成される。 ファイル パーミッションを制御するために umask を構成する方法の詳細については、「管理者ガイド」で「RSA RADIUS の管理」の章を参照してください。
RADIUS_HIGH_FDS	• この値を 0 に設定すると、ファイル ディスクリプタの管理が無効になる。ULIMIT_OPEN_FILES パラメータを 256 以下の値に設定した場合、RADIUS_HIGH_FDS を 0 に設定することができる。 • 1 に設定すると、ファイル ディスクリプタの管理が有効になる。ULIMIT_OPEN_FILES パラメータを 256 より大きい値に設定した場合は、RADIUS_HIGH_FDS を 1 に設定すること。 デフォルト値は 1。
ORACLE_MSB_FILE	ロケール固有の Oracle メッセージ ファイルへの絶対パスを指定する。 • メッセージ ファイルへのパス名を指定した場合、返されるファイル ディスクリプタは 255 より大きくなる。 • “” を指定した場合、RSA RADIUS は標準ライブラリの open() 呼び出しによって返されるディスクリプタを使用する。
RADIUS	デフォルト値は「radius」。 テクニカル サポートから指示があった場合以外は、この値を変更しないこと。
RADIUSOPTS	RSA RADIUS の実行時に使用するオプションを指定する。 デフォルト値は “”。 テクニカル サポートから指示があった場合以外は、この値を変更しないこと。
RADIUSARGS	デフォルト値は「sbr.xml」。 テクニカル サポートから指示があった場合以外は、この値を変更しないこと。

パラメータ	機能
RADIUS_PRIVATE_DIR	デフォルト値は「$RADIUSDIR」。 テクニカル サポートから指示があった場合以外は、この値を変更しないこと。
WATCHDOGENABLE	• この値を 0 に設定すると、RSA RADIUS の実行が失敗したときに RSA RADIUS を再起動する RSA RADIUS watchdog プロセスが無効になる。 • この値を 1 に設定すると、RSA RADIUS watchdog プロセスが有効になる。 デフォルト値は 0。
WATCHDOG	RSA RADIUS watchdog プロセスの名前を指定する。 デフォルト値は radiusd。 テクニカル サポートから指示があった場合以外は、この値を変更しないこと。
WATCHDOGOPTS	デフォルト値は --config $RADIUSDIR/radiusd.conf --pidfile $RADIUSDIR/radius.pid。 テクニカル サポートから指示があった場合以外は、この値を変更しないこと。
WATCHDOGARGS	デフォルト値は $RADIUSDIR/$SELF。 テクニカル サポートから指示があった場合以外は、この値を変更しないこと。

3 securid.ini ファイル

securid.ini ファイルを使用すると、RSA SecurID 認証で使用するデフォルト プロンプト文字列をカスタム文字列に置き換えることができます。デフォルト プロンプト文字列が長すぎて表示できない場合は、カスタム プロンプト文字列を使用すると便利です。

注：初期設定ファイルには、現在のリリースの RSA RADIUS で使用しないパラメータやセクションが含まれていることがあります。本書では、現在のリリースに関連しているセクションおよびパラメータについてのみ説明しています。本書で説明していないパラメータは変更しないでください。

RSA RADIUS は、デフォルト プロンプト文字列ではなく、**securid.ini** ファイルで指定されているプロンプト文字列を使用します。文字列のセットは、全体を置き換えることも、部分的に置き換えることもできます。**securid.ini** ファイル内に対応するエントリーが存在しないプロンプト文字列については、デフォルト プロンプト文字列が使用されます。

［Configuration］セクション

securid.ini の［Configuration］セクションでは、RSA SecurID のアクセス設定を指定します。

```
[Configuration]
Enable = 1
AllowSystemPins = 0
CheckUserAllowedByClient = 1
DefaultProfile = DEFAULT
```

次の表は、［Configuration］のパラメータとそれらの機能を示しています。

パラメータ	機能
Enable	• 1 に設定した場合、RSA RADIUS は RSA SecurID を使用してユーザーを認証することができる。 • 0 に設定した場合、RSA RADIUS は RSA SecurID を使用してユーザーを認証できない。 デフォルト値は 1。
AllowSystemPins	• 1 に設定すると、New PIN モード時にシステム生成 PIN を受け取るように RSA Authentication Manager で構成されているユーザーが受け付けられる。 • 0 に設定すると、New PIN モード時にシステム生成 PIN を受け取るように RSA Authentication Manager で構成されているユーザーが拒否される。 デフォルト値は 0。
CheckUserAllowed ByClient	• 1 に設定すると、RADIUS サーバはユーザーがネットワーク アクセス デバイスを通じた接続を許可されているかどうかを確認する。 • 0 に設定すると、RADIUS サーバはユーザーがネットワーク アクセス デバイスを通じた接続を許可されているかどうかを確認しない。 デフォルト値は 1。 **注：**このパラメータを 1 に設定する場合、RADIUS クライアントは Authentication Manager 内でエージェント ホストとして構成されている必要があります。 詳細については、「管理者ガイド」で「RSA RADIUS の管理」の章の「RADIUS クライアントの管理」を参照してください。

［Server_Settings］セクション

securid.ini の［Server_Settings］セクションでは、拡張ワンタイム パスワード（EOTP/EAP-15）認証または保護ワンタイム パスワード（POTP/EAP-32）認証の設定を指定します。

```
[Server_Settings]
Greeting =
Return_MPPE_Keys = 1
```

次の表は、［Server_Settings］のパラメータとそれらの機能を示しています。

パラメータ	機能
Greeting	ユーザーの認証後にネットワーク アクセス デバイスに返す 80 文字までの文字列。「Welcome to RSA Software」など。
Return_MPPE_Keys	0 に設定すると、モジュールは RADIUS MS-MPPE Send-Key 属性および MS-MPPE-Recv-Key 属性を転送しない。 1 に設定すると、モジュールはアクセス ポイントに送信する最後の RADIUS Access-Accept 応答に RADIUS MS-MPPE Send-Key 属性および MS-MPPE-Recv-Key 属性を含める。これは、アクセス ポイントが WEP 暗号化キーを生成するために必要となる。 アクセス ポイントで認証される対象がエンド ユーザーのみであり、WEP が使用されない場合は、この属性を 0 に設定することができる。 デフォルト値は 1。

［Prompts］セクション

RSA RADIUS は、デフォルト プロンプト文字列ではなく、**securid.ini** ファイルで指定されているプロンプト文字列を使用します。文字列のセットは、全体を置き換えることも、部分的に置き換えることもできます。**securid.ini** ファイル内に対応するエントリーが存在しないプロンプト文字列については、デフォルト プロンプト文字列が使用されます。

代替文字列の形式

代替文字列の中で変数テキストの位置を指定するには、%s を使用します。文字列に含まれる %s プレースホルダの数は、0 個、1 個、2 個のいずれかです。独自のプロンプト文字列を作成するときは、あらかじめ決められた数の %s プレースホルダを文字列に含める必要があります。文字列名には、確認のため、%s プレースホルダの数を反映するサフィックスが付加されています。

- 2 個の %s プレースホルダを必要とする文字列の名前には、_S_S というサフィックスが付加されます。通常、最初の %s プレースホルダは番号の範囲（「4 ～ 8」）を表します。2 番目の %s プレースホルダには、「Characters」または「Digits」が表示されます（または Characters 設定および Digits 設定で指定した、これらに相当する単語）。
- 1 個の %s プレースホルダを必要とする文字列の名前には、_S というサフィックスが付加されます。この %s プレースホルダは、システム生成 PIN に置換されます。
- %s プレースホルダを必要としない文字列の名前には、サフィックスが付加されません。

securid.ini ファイルに誤った形式の文字列が含まれている場合、その文字列は無視され、デフォルト プロンプト文字列が使用されます。

次の表は、**securid.ini** ファイルで使用する形式の表記法を示しています。

規則	説明
\b	バックスペース。通常は使用しない。
\f	改ページ文字
\n	改行文字。通常は \r と組み合わせて使用する。
\r	復帰文字。通常は \n と組み合わせて使用する。
\t	水平タブ
\v	垂直タブ。通常は使用しない。
\\	画面に表示されるバックスラッシュまたは円記号
\’	画面に表示される一重引用符文字
\”	画面に表示される二重引用符文字

代替文字列内で前述以外の文字の前にバックスラッシュまたは円記号がある場合、バックスラッシュまたは円記号は無視され、その文字は変更されずに表示されます。

引用符で囲んだ文字列

末尾にスペースのある文字列を引用符で囲んでいない場合は、RSA RADIUS が読み込む際に末尾のスペースが無視されます。末尾にスペースのある代替文字列を指定するには、文字列の先頭と末尾に二重引用符を挿入して、スペースを含む文字列全体を囲みます。たとえば、PIN という単語の後ろにコロンと 1 つのスペースが表示されるように文字列を指定するには、**StringName="PIN: "** と入力します（コロンと末尾の二重引用符の間にスペースを挟みます）。

例 1：詳細な代替文字列

次の図は、デフォルト プロンプト文字列のリストです。この図には、1 つのエントリーが 2 行に折り返し表示されている場合がありますが、実際に **securid.ini** を記述する場合は、各エントリーを 1 行に記述する必要があります。

```
;[Prompts]
;InputNextCode         = \r\nPlease Enter the Next Code from Your Token:
;InputMutChoose_S_S = \r\n  Enter your new PIN, containing %s %s,\r\n
or\r\n   <Ctrl-D> to cancel the New PIN procedure:
;InputCannotChoose     = \r\n   Press <Return> to generate a new PIN and
display it on the screen,\r\n          or\r\n   <Ctrl-D> to cancel the New
PIN procedure:
;InputMayChoose_S_S  = \r\n  Enter your new PIN, containing %s %s,\r\n
or\r\n   Press <Return> to generate a new PIN and display it on the
screen,\r\n           or\r\n   <Ctrl-D> to cancel the New PIN procedure:
;InputReadyForPin      = \r\n\r\nARE YOU PREPARED TO HAVE THE SYSTEM
GENERATE A PIN?(y or n) [n]:
;InputReadyForPin_1_S = \r\n\r\nPIN:         %s\r\n\r\n  10 second display
or Hit RETURN to continue.
;InputReenterPin       = \r\n                  Please re-enter new PIN:
;InputReenterPin_1     = \r\nPINs do not match.Please try again.\r\n
;OutputReject        = \r\n\r\nPIN rejected.Please try again.\r\n\r\nEnter
PASSCODE:
;OutputChange          = \r\n\r\nWait for the code on your card to change,
then log in with the new PIN\r\n\r\nEnter PASSCODE:
;OutputAccepted        = \r\nPASSCODE Accepted\r\n
;OutputDenied          = \r\nAccess Denied\r\n\r\n\r\nEnter PASSCODE:
;OutputNoPassReqd      = \r\nPASSCODE Not Required\r\n
;OutputDeniedFinal     = \r\nAccess Denied\r\n\r\n
;Characters            = characters
;Digits                = digits
```

例 2：2×40 ディスプレイ用の代替文字列

次の図は、2 行 × 40 文字のディスプレイ用に作成されたプロンプト文字列を示しています。この図には、1 つのエントリーが 2 行に折り返し表示されている場合がありますが、実際に **securid.ini** を記述する場合は、各エントリーを 1 行に記述する必要があります。

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;
; BEGINNING OF 2 lines by 40 characters prompts, these use the full 40
; character width (not including "\r\n") and one or two lines
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;
;[Prompts]
;InputNextCode          = Please Enter the Next Code from\r\nYour Token
;InputMustChoose_S_S    = Enter your new PIN (%s %s)
;InputCannotChoose     = Press <Return> to generate a new PIN and\r\ndisplay
it
;InputMayChoose_S_S     = Enter new PIN (%s %s) or press\r\n<Return> to
generate a new one
;InputReadyForPin       = ARE YOU PREPARED TO HAVE THE SYSTEM\r\nGENERATE A
PIN?(y or n) [n]
;InputReadyForPin_1_S = PIN:%s, 10 second display or\r\npress <Return> to
continue
;InputReenterPin        = Please re-enter new PIN
;InputReenterPin_1      = PINs do not match,\r\nPlease try again
;OutputReject           = PIN Rejected, please try again\r\nEnter PASSCODE
;OutputChange           = Wait for the code on your card to change\r\n then
log in with new PIN, Enter PASSCODE
;OutputAccepted         = PASSCODE Accepted
;OutputDenied           = Access Denied\r\nEnter PASSCODE
;OutputNoPassReqd       = PASSCODE Not Required
;OutputDeniedFinal      = Access Denied
;Characters             = chars
;Digits                 = digits
```

例 3：簡潔な代替文字列

次の図は、ユーザー用ではなく、クライアント エンドポイントのプログラムによる解析用に作成したプロンプト文字列を示しています。

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;
; BEGINNING OF extremely terse prompts.These are appropriate for automatic
; interpretation by another program which parses the prompts. A well trained
; end user could use these.
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;
;[Prompts]
;InputNextCode         = Next code
;InputMustChoose_S_S   = Must choose
;InputCannotChoose     = Cannot choose
;InputMayChoose_S_S    = May choose (%s, %s)
;InputReadyForPin      = Ready for pin
;InputReadyForPin_1_S  = Ready for pin 1
;InputReenterPin       = Reenter pin
;InputReenterPin_1     = Reenter pin 1
;OutputReject          = Reject
;OutputChange          = Change
;OutputAccepted        = Accepted
;OutputDenied          = Denied
;OutputNoPassReqd      = No pass reqd
;OutputDeniedFinal     = Denied final
;Characters            = chars
;Digits                = digits
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;
```

4 属性処理ファイル

この章では、RSA RADIUS 属性を指定するための、RADIUS 属性処理ファイルおよび Dictionary ファイルの使用法と設定について説明します。

注：属性処理ファイルおよび Dictionary ファイルには、本リリースの RSA RADIUS では使用されないパラメータやセクションが含まれている可能性があります。本書では、現在のリリースに関連しているセクションおよびパラメータについてのみ説明しています。本書で説明していないパラメータは変更しないでください。

注：.ini ファイル内の設定を編集する場合は、標準的な .ini 構文に準拠する必要があります。.ini ファイルに対して何らかの変更を加えた後は、それらの変更を反映するために RADIUS サーバを再起動する必要があります。

概要

vendor.ini ファイル内に記載されている各製品について、RSA RADIUS では Dictionary（.dct）ファイルを提供しています。この Dictionary ファイルは、RSA RADIUS と RADIUS クライアント間の属性交換に使用されます。初期化ファイルと同様、Dictionary ファイルは起動時にロードされます。

- Dictionary ファイルには、特定のタイプのデバイスから RSA RADIUS に RADIUS 要求を送信する際、RADIUS 要求内に含まれることが想定される属性を指定します。
- Dictionary ファイルには、RSA RADIUS から特定のタイプのデバイスに RADIUS 応答を送信する際、RADIUS 応答内に含める必要がある属性を指定します。

次の図は、サンプル Dictionary ファイルの形式を示しています。

```
##############################################################
#####
# Juniper.dct - RADIUS dictionary for Juniper M-160 and
M-40Es

# (See README.DCT for more details on the format of this
file)
##############################################################
#####
# Use the RADIUS specification attributes
#
@radius.dct

#
# Juniper specific parameters
#
MACRO Juniper-VSA(t,s) 26 [vid=2636 type1=%t% len1=+2
data=%s%]

ATTRIBUTE Juniper-Local-User-Name       Juniper-VSA(1,
string) r
ATTRIBUTE Juniper-Allow-Commands        Juniper-VSA(2,
```

Dictionary ファイル レコード

Dictionary ファイル内のレコードは、次の表に示すいずれかのキーワードで開始する必要があります。

キーワード	機能
@	参照先ファイルをインクルード
ATTRIBUTE	新しい属性を定義
VALUE	属性用の名前付き整数値を定義
MACRO	反復定義を容易にするためのマクロを定義
OPTIONS	属性定義の範囲を越える補足情報を定義
#	コメント（テキストは無視される）

Dictionary ファイルの編集

RSA RADIUS に付属する製品固有のファイルには、ベンダー固有の RADIUS クライアント実装が反映されています。したがって、通常は付属の Dictionary ファイルを修正する必要はありません。ただし、ネットワーク アクセス デバイスのベンダーから新製品や新しい属性 / 属性値に関する情報が提供された場合は、Dictionary ファイルを編集することにより、既存の RSA RADIUS 構成にこれらの情報を追加できます。

Operations Console を使用して、新しい Dictionary ファイルをアップロードし、2 つの RADIUS 構成ファイル（**vendor.ini** および **dictiona.dcm**）を編集します。詳細については、次の Operations Console のヘルプ トピックを参照してください。

- RADIUS ディクショナリの追加
- ディクショナリへの RADIUS 属性定義の追加
- ディクショナリの RADIUS 属性定義の変更

インクルード レコード

Dictionary ファイル内の先頭が @ 文字のレコードは、特殊なインクルード レコードとして処理されます。@ 文字の後に続く文字列は、インクルード対象の Dictionary ファイル名を示します。たとえば、@vendorA.dct エントリーは、**vendorA.dct** ファイル内のすべてのエントリーをインクルードする必要があることを示します。

インクルード レコードが参照される深さは 1 階層のみです。たとえば、**vendorA.dct** に **radbase.dct** ファイルがインクルードされ、**radbase.dct** に **radacct.dct** がインクルードされている場合、**vendorA.dct** には、**radbase.dct** のレコードのみが組み込まれ、**radacct.dct** のレコードは組み込まれません。

マスター Dictionary ファイル

マスター Dictionary ファイル **dictiona.dcm** は、ベンダー固有の Dictionary を参照するインクルード レコードで構成されています。ベンダー固有の Dictionary をマスター Dictionary にインクルードする順番は、2 つのベンダー固有の Dictionary 内に同一の属性または属性値に対する異なる定義が含まれている場合にのみ注意が必要です。最初にインクルードされた属性または属性値の定義が、同じ属性または属性値に対する後続の定義よりも優先されます。たとえば、マスター Dictionary ファイル **dictiona.dcm** が、次のインクルード レコードで構成されているとします。

@vendorA.dct
@vendorB.dct
@vendorC.dct

この場合、**vendorA.dct** で定義されている属性および属性値は、**vendorB.dct** または **vendorC.dct** で定義されている属性および属性値より優先され、**vendorB.dct** で定義されている属性および属性値は **vendorC.dct** で定義されている属性および属性値より優先されます。

属性レコード

属性レコードは、次の構文に従います。

```
ATTRIBUTE 属性名 属性 ID 構文型 フラグ
```

パラメータ	機能
属性名	属性の名前（最大 31 文字、空白を含めることはできない）。
属性 ID	属性のエンコードされた RADIUS 識別子を示す整数値（0 ～ 255）。
構文型	属性の構文型。
フラグ	属性をチェックリストまたはリターンリスト（またはその両方）に含めるかどうか、属性が複数値を持つかどうか、および属性が順序指定可能かどうかを定義。

注：標準の Dictionary ファイルに対する制限のうち、すべての属性レコードの属性 ID が一意でなければならないという制限は、マスター Dictionary ファイルには適用されません。同一の属性 ID に対して、複数のベンダーが異なる属性名を定義している場合もあります（その属性 ID が RADIUS 標準仕様でまだ使われていないことを想定しています）。RSA RADIUS データベース内の属性は（属性 ID ではなく）名前によって保存されるため、この状況により、データベース内の情報に混乱が生じることはありません。

次の例は、典型的な属性レコードを示しています。

```
ATTRIBUTE Framed-IP-Netmask 9 ipaddr Cr
```

この属性レコードには、次に示すすべての情報が定義されています。

- 属性名「Framed-IP-Netmask」がサポートされる。
- 属性のエンコードされた RADIUS ID は 9。
- この属性は IP アドレス構文を使用する必要がある。
- フラグ文字は、属性がチェックリスト内に複数回出現でき（C）、RSA RADIUS データベース内のユーザー / プロファイル エントリー用のリターン リスト内に 1 回のみ出現できる（r）ことを示す。

属性名と属性 ID

1 つの Dictionary ファイル内の 2 つの属性レコードに、同一の属性名または属性 ID を指定することはできません。属性名または属性 ID が重複している場合は、最初の属性定義が優先され、後続の定義は無視されます。

構文型識別子

次の表は、標準的な構文型識別子の一覧を示しています。

構文型	機能
hexadecimal	16 進数文字列。
hex4	4 バイト（32 ビット）の符号なし 16 進数値。
int1	1 バイト（8 ビット）の符号なし 10 進数値。
int4、integer	4 バイト（32 ビット）の符号なし 10 進数値。
signed-integer	4 バイト（32 ビット）の符号付き 10 進数値。先頭ビットが 1 の数字は負数と解釈される。
ipaddr	IP アドレス属性または IP ネットマスク属性。
string	（ヌル終端文字を含む）文字列属性。
stringnz	（ヌル終端文字を含まない）文字列属性。
time	時間属性。00:00:00 GMT（1970 年 1 月 1 日）からの経過秒数で指定。

注：符号付き整数は、パケットの形で受信される属性およびそれらの属性に関連する処理（アカウンティング ログ、認証ログ、認証レポート、SQL プラグ インなど）でのみサポートされます。

複合構文型

前述の表に示した標準の構文型識別子に加えて、Dictionary ファイルはベンダー固有の属性定義に使用される複合構文型にも対応できます。複合構文型を使用する場合は、1 つの構文型識別子を使用する代わりに、次に示す 1 つ以上のオプションを組み合わせて角括弧内に記述します。

オプション	機能
vid=*nnn*	（ISO により割り当てられる）デバイス メーカーの SMI ネットワーク管理プライベート エンタプライズ コード（10 進数形式）。
type*N*=*nnn*	RADIUS 仕様に定義されたベンダー固有の属性タイプ設定。*N* はフィールド長（バイト単位）を、*nnn* は 10 進数形式のフィールド値を示す。
len*N*=*nnn*	RADIUS 仕様に定義されたベンダー固有の属性の長さフィールド。*N* はフィールド長（バイト単位）を、*nnn* は 10 進数形式のフィールド値を示す（値の先頭が + 符号の場合は、データ部分の長さを *nnn* に加算することにより、実際の長さが得られる）。

オプション	機能
data= 構文型	属性内に含める実データ。任意の標準構文型を使用可能。
tag=*nnn*	トンネル属性には、同一トンネルを参照する属性を同一パケット内でグループ化するためのタグ フィールドが含まれる。ベンダーの機器によってはタグをサポートしていないため、この構文型は省略可能であり、属性にタグ フィールドを含める場合にのみ必要。 値 0 は、フィールドが存在する必要があるが無視されることを示す。

ベンダー固有の属性（VSA）の定義例を次に示します。

```
ATTRIBUTE vsa-xxx 26 [vid=1234 type1=1 len1=+2 data=string] R
```

フラグ文字

フラグ設定は、次の表に示されている 1 つ以上のフラグ文字を連結したもので構成されます。

フラグ文字	意味
b または B	特定ベンダー ID 用の 1 つのベンダー固有の属性内に属性をバンドルできることを示す。一連のサブ属性の 1 つとして 1 つの VSA 内にインクルードされる可能性がある。
c	属性はユーザーまたはプロファイルのチェックリスト内に 1 回のみ出現可能。
C	属性はユーザーまたはプロファイルのチェックリスト内に複数回出現可能。
r	属性はユーザーまたはプロファイルのリターンリスト内に 1 回のみ出現可能。
R	属性はユーザーまたはプロファイルのリターンリスト内に複数回出現可能。
t	属性はトンネル属性リスト内に 1 回のみ出現可能。
T	属性はトンネル属性リスト内に複数回出現可能。
o または O	属性の順序指定が可能。このタイプの属性について、管理者は RSA RADIUS データベース内に属性を保存する順番を制御できる。このフラグは複数値をとる属性の場合にのみ意味を持つ。

値レコード

値レコードは、事前に定義した整数属性の特定整数値の名前を定義するために使用します。値レコードは必須ではありませんが、属性の整数値に特定の意味を持たせる場合は便利です。値レコードは次の構文に従う必要があります。

```
VALUE 属性名 値名 整数値
```

パラメータ	機能
属性名	属性の名前（最大 31 文字、空白を含めることはできない）
値名	属性値の名前（最大 31 文字、空白を含めることはできない）
整数値	属性値に関連づけられる整数値

Dictionary ファイル内の 2 つの値レコードに、同一の属性名と値名、または同一の属性名と整数値を指定することはできません。定義が重複している場合は、最初の属性定義が優先されて後続の定義は無視されます。

次の例では、値レコードを使用して、Framed-Protocol 属性の値に対してユーザーが判別しやすい名前を付けています。

```
ATTRIBUTE   Framed-Protocol     7     integer     Cr
VALUE       Framed-Protocol     PPP           1
VALUE       Framed-Protocol     SLIP          2
```

これらの Dictionary レコードを使用することで、管理者は Framed-Protocol 属性を使用するときに、整数値 1 が PPP を、整数値 2 が SLIP を意味することを覚えておく必要がなくなります。その代わりに、Security Console 上で属性値の一覧から目的の値を選択できます。

マクロ レコード

マクロ レコードは、多くの共通パラメータを含んでいる複数のベンダー固有の属性を効率よく作成するために使用されます。マクロ レコードを使用すると、レコード内の共通部分をカプセル化できます。マクロ レコードは次の構文に従う必要があります。

```
MACRO マクロ名 ( マクロ変数 ) 代替文字列
```

パラメータ	機能
マクロ名	マクロの名前
マクロ変数	カンマで区切られた 1 つ以上のマクロ変数名
代替文字列	マクロ変数に代入される文字列。%x% 形式に従っているすべての文字シーケンスについて代入処理が実行され、マクロ変数 *x* に代入される

次の例では、複数のベンダー固有の属性を効率よく指定するためにマクロを使用しています。

```
MACRO Cisco-VSA(t, s) 26 [vid=9 type1=%t% len1=+2 data=%s%]
ATTRIBUTE   Cisco-xxx   Cisco-VSA(1, string)    R
ATTRIBUTE   Cisco-yyy   Cisco-VSA(4, int4)      C
ATTRIBUTE   Cisco-zzz   Cisco-VSA(9, ipaddr)    r
```

RSA RADIUS の Dictionary 処理に組み込まれたマクロ プリプロセッサにより、前述の例のレコードは次のように変換されます。

```
ATTRIBUTE Cisco-xxx 26 [vid=9 type1=1 len1=+2 data=string] R
ATTRIBUTE Cisco-yyy 26 [vid=9 type1=4 len1=+2 data=int4]   C
ATTRIBUTE Cisco-zzz 26 [vid=9 type1=9 len1=+2 data=ipaddr] r
```

オプション レコード

デフォルトでは、各ベンダー固有の属性は単一の VSA 属性内にエンコードされます。VSA 属性の形式は、次の表に示すとおりです。

ビット	内容
0 ～ 7	タイプ：値 26 が入る
8 ～ 16	データ長（バイト単位）
17 ～ 47	ベンダー ID
48 以降	ベンダー データ

ただし、オプション設定にパラメータを記述することにより、1 つの VSA レコードのベンダー データ部分に複数のベンダー固有の属性を格納することも可能です。

オプション レコードは次の形式に従う必要があります。

```
OPTION bundle-vendor-id = vid
```

注：属性をバンドルするには B フラグを設定しなければなりません。特定のベンダー固有の属性をバンドルするためには、そのベンダーのベンダー ID を指定したオプション レコードを設定し、目的の属性に B（または b）フラグを設定する必要があります。

Nortel Rapport Dictionary はこのオプションをサポートしています。Nortel のベンダー固有の属性を 1 つの VSA 内にバンドルするには、次のエントリーを記述します。

```
OPTION bundle-vendor-id=562
```

マクロ レコードに設定されているとおり、562 は Nortel のベンダー ID です。このように指定することで、Nortel Rapport の複数のベンダー固有の属性は、RADIUS VSA 属性のベンダー データ部分に連結されます（最大 249 オクテット）。

classmap.ini ファイル

classmap.ini 初期設定ファイルには、受信したアカウンティング要求内に含まれる 1 つ以上の Class 属性内にエンコードされている RADIUS 属性に対して、RSA RADIUS が実行すべき動作が指定されています。

［AttributeName］セクション

classmap.ini ファイルの［AttributeName］セクションには、Class 属性内にカプセル化されている RADIUS 情報をアカウンティング要求に追加するか、アカウンティング要求内の現在値を置き換えるかを指定します。ある属性を別の属性で置き換える場合は、別の識別子を使用して元の属性を要求に追加できます。

```
[AttributeName]
<add | replace> = Attribute [,Attribute]
```

次の表は、［AttributeName］セクションのパラメータとその機能を示しています。

パラメータ	機能
AttributeName	認証サーバによって Class 属性内にエンコードされた属性の名前。
<add | replace>	（他の属性値に影響を与えることなく）属性値をアカウンティング要求に追加するか、アカウンティング要求内の値を置き換えるかを指定する。
Attribute	アカウンティング要求に追加する属性の名前を指定する。この属性には、*AttributeName* で指定した属性の元の値が含まれる。
[,*Attribute*]	*AttributeName* の値で既存の *Attribute* 値を置き換えるときに、置き換える属性の値を格納するアカウンティング要求内の属性名を指定する。 replace キーワードが使用されている場合のみ有効。

注：RADIUS Class 属性に IPv6 属性を含めることはできません。

次の例では、カプセル化された User-Name 属性で、アカウンティング要求内の既存の User-Name を置き換えています。

```
[User-name]
replace = User-Name
```

次の例では、カプセル化された User-Name 属性を User-Name としてアカウンティング要求内に配置し、User-Name の元の値は Funk-Full-User-Name として要求に追加しています。

```
[User-name]
replace = User-Name, Funk-Full-User-Name
```

次の例では、カプセル化された User-Name 属性を新しい属性としてアカウンティング要求に追加しています。元の User-Name 属性はそのまま保持されます。

```
[User-Name]
add = Funk-Full-User-Name
```

filter.ini ファイル

注：filter.ini ファイルの設定には Operations Console を使用します。

filter.ini 構成ファイルには、着信 / 発信される RADIUS パケットに対する属性フィルタリング ルールを設定します。

フィルタ ルール

filter.ini ファイル内の各フィルタは、角括弧で囲まれたフィルタ名（[名前]）とそのフィルタに対するルールで構成されます。

各ルールは、次の 3 種類のいずれかの形式に従います。

キーワード 属性値
キーワード 属性
キーワード

次の表は有効な構文の組み合わせを示しています。

filter.ini ルール構文	機能
ALLOW	このキーワードをパラメータなしに使用した場合は、すべての属性を（値とは無関係に）パケットに含めることが許可される。
ALLOW 属性	指定した属性を（値とは無関係に）パケットに含めることが許可される。
ALLOW 属性 値	パケット内で許可される特定の属性 / 値の組み合わせを指定する。
EXCLUDE	このキーワードをパラメータなしに使用した場合は、すべての属性が（値とは無関係に）パケットから除外される。 EXCLUDE はフィルタのデフォルト動作。
EXCLUDE 属性	指定した属性が（値とは無関係に）パケットから除外される。
EXCLUDE 属性 値	パケットから除外する属性 / 値の組み合わせを指定する。

filter.ini ルール構文	機能
ADD 属性 値	パケット内に追加する特定の属性 / 値の組み合わせを指定する。この属性は、その他のすべてのルールが処理された後で追加される。
REPLACE 属性*1* WITH 属性*2*	出現するすべての属性*1*を、属性*1*の値を保持したままで、属性*2*に置き換える。
REPLACE 属性*1* WITH 属性*2* 値*2*	出現するすべての属性*1*を（値とは無関係に）属性*2*に置き換える。属性値には値*2*が使用される。
REPLACE 属性*1* 値*1* WITH 属性*2*	出現するすべての属性*1*（属性値は値*1*）を、値*1*の値を保持したままで、属性*2*に置き換える。
REPLACE 属性*1* 値*1* WITH 属性*2* 値*2*	出現するすべての属性*1*（属性値は値*1*）を、属性値が値*2*の属性*2*で置き換える。

ルールに適合する属性のみがパケットに追加されます。属性には、RADIUS パケット内に 1 回のみ出現できるものと、複数回出現できるものがあります。ADD ルールに指定された属性が、（ALLOW ルールおよび EXCLUDE ルールの適用後に）パケット内にすでに存在しており、その属性が 1 回のみ出現可能である場合には、ADD ルールは適用されず 2 番目の属性インスタンスは追加されません。

RSA RADIUS の Dictionary ファイル **radius.dct** には、RADIUS 標準に定義された特定の整数値に対する文字列エイリアスが含まれます。これらの文字列は、属性フィルタ ルール内で使用可能です。

注：フィルタ ルールの柔軟性は非常に優れています。ただし、RSA RADIUS には、無効な RADIUS パケットの作成を防止する機能はありません。要求のタイプによっては使用に適さない属性もあります。たとえば、プールされた Framed-Ip-Address 属性をアカウンティング要求に追加すると、使用可能な IP アドレスがなくなる可能性があります。

フィルタ ルールの順番

ルールの順番は重要です。パラメータのない汎用的な性質のデフォルト ルール、たとえば ALLOW（別途指定されない限りはすべての属性を許可）や EXCLUDE（別途指定されない限りはすべての属性を除外）などは、フィルタ内の先頭に指定する必要があります。後から記述されたルールほど優先され、最後に適用可能なルールが「勝利」します。ADD ルールおよび REPLACE ルールは、ALLOW ルールおよび EXCLUDE ルールの後で適用されます。

より多くのパラメータを含んだより限定的なルール（ADD 属性 値など）は、より少ないパラメータを含むより汎用的なルール（ALLOW 属性、EXCLUDE など）に対する例外処理として機能します。たとえば、特定の属性を許可（ALLOW）し、その属性の特定の値を除外（EXCLUDE）するといったケースです。または、すべての属性を除外（EXCLUDE）し、特定の属性のみ許可（ALLOW）し、さらに特定の属性 / 値の組み合わせを追加（ADD）することも可能です。

フィルタ設計時には、次のいずれかの手法を使用します。

- ルール リストの先頭にデフォルトの EXCLUDE ルール（パラメータなし）を指定し、次に、パケット内に含める必要がある属性または属性 / 値の組み合わせに対する ALLOW ルールを追加します。ADD ルールおよび REPLACE ルールも使用可能です。
- ルール リストの先頭にデフォルトの ALLOW ルール（パラメータなし）を指定し、パケットから除外する必要がある属性または属性 / 値の組み合わせに対する EXCLUDE ルールを追加します。ADD ルールおよび REPLACE ルールも使用可能です。

filter.ini のデフォルト動作は EXCLUDE です。一切のルールを含まないフィルタを適用すると、パケット内のすべての属性が削除されます。

フィルタ ルール内の値

属性の値は、属性 Dictionary 内の属性型に基づいて解釈されます。次の表は、各属性型の意味を示しています。

属性型	機能
hexadecimal	16 進数値は文字列として指定される。エスケープ コードを使用して特殊文字を含めることも可能。
int1、int4、integer	1 バイトまたは 4 バイトの符号なし 10 進数値（integer は int4 に同じ）。 **注：**RSA RADIUS の Dictionary ファイル **radius.dct** には、RADIUS 標準に定義された特定の整数値に対する文字列エイリアスが含まれます。これらの文字列は、属性フィルタ ルール内で使用可能です。
ipaddr、ipaddr-pool	ドット表記の IP アドレス。例： `EXCLUDE NAS-IP-Address 127.0.0.1`
ipxaddr-pool	一連の 16 進数値。例： `ALLOW Framed-IPX-Network 0042A36B`

属性型	機能

String	（ヌル終端文字を含む）文字列属性。文字列はテキストとして指定される。テキストを二重引用符（"）で囲むことも可能。テキストは正規表現として解釈される。バックスラッシュまたは円記号（\）はエスケープ文字と解釈される。エスケープコードの意味は次のとおり。
	コード　意味
	\a　7
	\b　8
	\f　12
	\n　10
	\r　13
	\t　9
	\v　11
	nnn　*nnn* は 0 ～ 255 の範囲の 10 進数値
	\x*nn*　*nn* は 00 ～ FF の範囲の 16 進数値
	c　*c* は文字どおりに解釈される 1 つの文字
	文字列内で通常の文字として使用するバックスラッシュまたは円記号（\）や、引用符で囲まれた文字列内の二重引用符（“）は、直前にエスケープ文字を付加しなければならない。次に例を挙げます。
	`ADD Reply-Message Session limit is one hour`
	`ADD Reply-Message "Session limit is one hour"`
	`ADD Reply-Message "Your username is \"George\""`

time	時間は日付と時刻を示す文字列で指定される。
	`yyyy/mm/dd hh:mm:ss`
	日付部分は必須。時刻部分は必要な正確性レベルに応じて指定でき、完全に省略することも可能。次に例を挙げます。
	`2006/4/3 14:00:00`
	`2006/4/3 14`
	は、いずれも 2006 年 4 月 3 日午後 2:00 を指す。
	次に例を挙げます。
	`ADD Ascend-PW-Expiration 2006/4/3`

属性フィルタの参照

RSA RADIUS の属性フィルタリングにより、パケット処理の柔軟性が向上します。導入環境でフィルタリングを無効にするには、*.pro、*.dir、**peapauth.aut**、**ttlsauth.aut** ファイルからフィルタリング パラメータを削除します。多くの場合、フィルタリングは、導入環境から「外部」に発信されるパケットに対してのみ適用されます（FilterOut パラメータを使用）。

プロキシまたはダイレクト導入環境の構成で **filter.ini** ファイル内に定義されているフィルタリング ルールを参照するには、［Auth］セクションおよび［Acct］セクション内の FilterOut パラメータと FilterIn パラメータを使用します。

完全な構文は、次のとおりです。

```
[Auth]
FilterIn= 名前 1
FilterOut= 名前 2

[Acct]
FilterIn= 名前 3
FilterOut= 名前 4
```

名前 *1*、名前 *2* などの部分はフィルタ名です。**filter.ini** ファイル内の各セクションを、［名前 *1*］、［名前 *2*］のようにフィルタ名として定義します。構文内の名前は、相互に完全に独立しています。すべてを同一の名前にすることも、すべてを異なる名前にすることも、一部の名前のみ同じにすることも可能です。

［Auth］セクションおよび［Acct］セクション内で FilterIn/FilterOut パラメータを指定するときは、角括弧なしのフィルタ名を記述するよう注意してください（「名前」のみを記述し、「［名前］」とはしません）。

注：filter.ini ファイル内に［名前］セクションが存在しない場合、すべての属性を除外（EXCLUDE）するフィルタを適用したときと同じ結果になります。言い換えると、存在しないフィルタ名を割り当てた場合には、最終的には一切の属性が含まれない空のパケットが生成されます。

注：アカウンティング フィルタ内で、RSA RADIUS IP アドレス プール内の IP アドレスは割り当てないでください。これらのアドレスが割り当てられると、解放されることはありません。

spi.ini ファイル

spi.ini 初期設定ファイル内には暗号化キーが定義されています。また、アカウンティング要求内の暗号化された Class 属性が、どのサーバから渡された属性なのかも指定されます。**spi.ini** ファイルを使用することで、ある RSA RADIUS サーバ上で認証されたセッションのアカウンティング要求を、別の RSA RADIUS サーバ上でデコードすることが可能になります。**spi.ini** に指定されていないサーバから渡された Class 属性は無視されます。

共通のネットワーク アクセス デバイスからの認証要求およびアカウンティング要求を受け取る可能性があるすべての RSA RADIUS サーバは、同様の **spi.ini** ファイルで構成する必要があります。この spi.ini ファイル内には、その「クラスタ」に属するすべてのサーバの IP アドレス一覧を記述しなければなりません。これにより、クラスタ内のあるサーバ上でユーザー認証と暗号化された Class 属性の生成を行い、クラスタ内の別のサーバ上でその属性を復号化して処理することが可能になります。

［Keys］セクション

spi.ini ファイル内の［Keys］セクションには、Class 属性内にカプセル化されるサブ属性をエンコードする暗号化キーの一覧を指定します。

```
[Keys]
CurrentKey = n
1 = 値
2 = 値
```

次の表は、［Keys］セクションのパラメータとその機能を示しています。

パラメータ	機能
CurrentKey	現在アクティブな暗号化キーを指定する。*n* 部分には 0、または［Keys］セクション内に記述したいずれかのキー番号を指定する。 0：Class 属性の暗号化用に一意のランダム キーを生成して使用する。このオプションは、RSA RADIUS サーバが他のサーバと暗号化された Class 属性を交換しない場合にのみ使用される。 *n*：Class 属性の暗号化に使用するキー番号を指定する。 デフォルト値は 0。
n = 値	暗号化キーの番号と値を指定する。

次の例では、RSA RADIUS サーバにより、Class 属性を暗号化する一意のランダム キーが生成されます。

```
[Keys]
CurrentKey = 0
```

次の例では、2 番目のキー（swordfish）が現在アクティブになっており、Class 属性の暗号化に使用されます。このセクション内のその他のキーは、同一クラスタ内の別のサーバから受信した Class 属性の復号化に使用できます。

```
[Keys]
CurrentKey = 2
1 = firstkey
2 = swordfish
3 = mypassword
```

［Hosts］セクション

spi.ini ファイル内の［Hosts］セクションには、どのサーバから渡された Class 属性について、カプセル化 / 暗号化されたサブ属性の解析を行うかを指定するため、そのサーバの IP アドレスを記述します。**spi.ini** 内の［Hosts］セクションに指定されていないサーバから渡された Class 属性に対しては、特別な処理は行われません。

［Hosts］セクション内の情報は、Class 属性内に含まれるサーバ識別子の算出に使用されます。［Hosts］セクション内にホストの 1 つのインタフェースのみが記述されている場合は、そのインタフェースがサーバ識別子の算出に使用されます。ホストの複数のインタフェースが記述されている場合は、リスト内の最後のインタフェースの IP アドレスが使用されます。一致するアドレスが見つからない場合は、ホストのプライマリ IP アドレスが使用されます。ホストのインタフェースのものではないアドレスは、Class 属性の送信元として受け入れられるその他一連のサーバの構成に使用されます。

次の例では、クラスタに属するサーバとして 3 つのサーバが定義されています。

```
[Hosts]
192.168.15.21
192.168.23.121
192.168.23.205
```

vendor.ini ファイル

vendor.ini 初期設定ファイル内には、RSA RADIUS が他社製品と連携して動作するために必要な情報が含まれています。

［Vendor-Product Identification］セクション

vendor.ini ファイルの［Vendor-Product Identification］セクションには、RSA RADIUS で使用できるネットワーク アクセス デバイスに関する情報が含まれています。

次の表は、［Vendor-Product Identification］セクションのパラメータとその機能を示しています。

パラメータ	機能
vendor-product	製品の名前を指定する。製品名は一意である必要があり、空白を含めることはできない（最大 31 文字）。これらの製品名は、［RADIUS Clients（RSA RADIUS クライアント）］パネルの［Maker/Model （製造元 / モデル）］リストでのみ使用される。このリストは、新しい RADIUS クライアントの追加時やベンダー固有の属性の選択時に使用される。
Dictionary	この製品に使用する Dictionary ファイルを指定する。Dictionary ファイルは、RSA RADIUS デーモン / サービスと同じディレクトリに配置する必要がある。Dictionary 名の拡張子の指定は不要（パラメータに指定した Dictionary 名に対して、拡張子 .DCT が自動的に付加される）。

パラメータ	機能
call-filter-attribute	フィルタ機能の呼び出し時に使用する属性を指定する。Ascend/Lucent ネットワーク アクセス デバイスでのみ使用される。
challenge-response-attribute	ネットワーク アクセス デバイスによるチャレンジ シーケンスへの応答送信時に使用される属性番号を指定する。 このパラメータを指定しない場合は、デフォルト動作として、User-Password 属性内に応答がエンコードされているものと想定される。
data-filter-attribute	データ フィルタ機能に使用する属性を指定する。Ascend/Lucent ネットワーク アクセス デバイスでのみ使用される。
discard-after	ユーザー名情報を「修飾された」形式で送信する着信プロキシ RADIUS サーバに対して使用する。たとえば、プロキシ RADIUS サーバから *username@company* 形式のユーザー名が送信される場合は、@ を指定することにより、認証時に区切り文字 @ およびそれ以降のテキストは破棄されて、文字列 *username* がユーザー名として使用される。
discard-before	ユーザー名情報を「修飾された」形式で送信する着信プロキシ RADIUS サーバに対して使用する。たとえば、プロキシ RADIUS サーバから *company$username* 形式のユーザー名が送信される場合は、$ を指定することにより、認証時に区切り文字 $ および $ より前のテキストは破棄されて、文字列 *username* がユーザー名として使用される。
help-id	ベンダー情報ヘルプ ファイル内のベンダー製品に関するヘルプ コンテキスト。
ignore-acct-ss	Yes に設定した場合は、共有シークレットによるアカウンティング パケットのデジタル署名は無視される。このオプションは、アカウンティング パケットに対する署名が適切に行われないデバイスへの対策として設けられている。 デフォルト値は No。
ignore-ports	あるユーザーに割り当てられたポートが現在別のユーザーによって使用されている場合に、RSA RADIUS が前のユーザーはログオフしたものとみなすかどうかを指定する。 • No に設定した場合は、前のユーザーはログオフしたものとみなされてアクティブ ユーザー リストから削除される。 • Yes に設定した場合は、この判断は行われず、両方のユーザーがアクティブとみなされる。 デフォルト値は No。

パラメータ	機能
max-eap-fragment	製造元 / モデルごとに最大 EAP フラグメント長を指定する。TLS または TTLS から送出される最大 EAP フラグメント長は、ここで設定した max-eap-fragment の値と、各 **.eap/.aut** ファイル内に指定されている最大値のうち、小さい方の値になる。 デフォルト値は 1020。ただし、効率性を考慮して、フラグメント長にはすべての顧客のアクセス ポイントに対応できるよう十分小さい値を指定する必要がある。
port-number-usage	• per-port-type に設定した場合は、アクティブ リスト内のエントリーのうち、重複するポート番号およびポート タイプを含んでいるエントリーは削除される。 • unique に設定した場合は、アクティブ リスト内のエントリーのうち、重複するポート番号を含んでいるエントリーが削除される（ポート タイプ情報は無視される）。 デフォルト値は per-port-type。
product-scan-acct	アカウンティング要求に関連づけられる製品を要求パケットの内容に基づいて動的に決定するためのルールを含んでいる **vendor.ini** ファイル内のセクション名を指定する。
product-scan-auth	認証要求に関連づけられる製品を要求パケットの内容に基づいて動的に決定するためのルールを含んでいる **vendor.ini** ファイル内のセクション名を指定する。
send-class-attribute	No に設定した場合は、Access-Accept 応答時に、クライアントに Class 属性が送信されない （この機能は Class 属性を適切に処理できないデバイスへの対策として設けられている）。 デフォルト値は Yes。
send-session-timeout-on-challenge	• Yes に設定した場合は、EAP メッセージを含んだ Access-Challenge 応答時に、Session-Timeout 属性がクライアントに送信される。この属性は、ネットワーク アクセス デバイスに対して、チャレンジに対するユーザー応答を待つべき時間を指定する。 • No に設定した場合は、EAP メッセージを含んだ Access-Challenge 応答時に、Session-Timeout 属性がクライアントに送信されない。 デフォルト値は Yes。

5 アカウンティング構成ファイル

この章では、サーバのアカウンティング機能を有効化、無効化、構成するための RSA RADIUS アカウンティング初期設定（.ini）ファイルの使用法と設定について説明します。account.ini ファイルは RSA RADIUS ディレクトリ内に存在し、起動時にロードされます。

注：初期設定ファイルには、現在のリリースの RSA RADIUS で使用しないパラメータやセクションが含まれていることがあります。本書では、現在のリリースに関連しているセクションおよびパラメータについてのみ説明しています。本書で説明していないパラメータは変更しないでください。

注：account.ini ファイル内の設定を編集する場合は、.ini 構文に準拠する必要があります。**account.ini** ファイルに何らかの変更を加えた後は、それらの変更を反映するために RADIUS サーバを再起動する必要があります。

account.ini ファイル

account.ini ファイルには、RSA RADIUS が RADIUS アカウンティング属性をどのようにカンマ区切りテキスト ファイルに記録するかを制御する情報が含まれます。**account.ini** ファイルでは、具体的にはファイル作成に関する設定（ファイル作成頻度、最大サイズ、デフォルト ディレクトリなど）とファイルの内容（受信したアカウンティング要求ごとに記録する情報など）を制御します。

［Alias/ 属性名］セクション

account.ini の［Alias/ 属性名］セクションは、名前は異なるが同じ意味を持つ属性を関連づけるために使用されます。たとえば、ある属性を Acct-Octet-Pkt と呼ぶネットワーク アクセス デバイス ベンダーがあれば、Acct-Oct-Packets と呼ぶベンダーもありますが、この 2 つの属性は同じことを意味します。

各［Alias/ 属性名］セクションでは、RSA RADIUS によってすでにログに記録されている 1 つの RADIUS アカウンティング属性を、任意の数の別の属性にマッピングすることができます。必要な数の［Alias/ 属性名］セクションを作成できます。各セクションで次の構文を使用する必要があります。

```
［Alias/ 属性名］
ベンダー固有属性 =
ベンダー固有属性 =
```

次の表は、［Alias/ 属性名］のパラメータとそれらの機能を示しています。

パラメータ	機能
属性名	優先して使用する属性名。属性名に指定する属性は、RSA RADIUS アカウンティング ログ ファイル（.act）の列に現在記録されている属性でなければならない。したがって、**account.ini** の［Attributes］セクションに含まれている属性の名前を指定する必要がある。
ベンダー固有属性	エントリーを 1 行に 1 つずつ入力する。ベンダー固有の属性名の直後にスペースを挟まずに等号（=）を入力する必要がある。不正な書式のエントリーは、無効なエントリーとみなされ無視される。

リストに指定した各ベンダー固有属性は、アカウンティング ログ ファイルのマッピングした属性名の列に記録されます。［Alias/ 属性名］セクションに指定するベンダー固有属性は、［Attributes］セクションに含まれていないことを確認してください。これらの属性の名前が［Attributes］セクションに含まれていると、その属性の列とマッピングした属性名の列の両方に記録されることになります。

［Alias/ 属性名］セクションで参照するすべての属性名は、Dictionary ファイルに定義し、予め RSA RADIUS サーバにインストールしておく必要があります。この条件は、属性名と各ベンダー固有属性の両方に該当します。

次の例では、標準の RADIUS 属性 Acct-Octet-Packets に、ベンダー固有属性 Acct-Octet-Pkt と Acct-Oct-Packets をマッピングしています。これら 3 つの属性のいずれについても、検出された値がアカウンティング ログ ファイルの Acct-Octet-Packets 列に記録されます。

```
[Alias/Acct-Octet-Packets]
Acct-Octet-Pkt=
Acct-Oct-Packets=
```

［Attributes］セクション

account.ini ファイルの［Attributes］セクションには、アカウンティング要求を受け取るたびにアカウンティング ログ ファイルに記録されるすべての属性のリストが含まれます。RSA RADIUS のインストール時に、**account.ini** ファイルには、すべての標準 RADIUS 属性とすべてのサポート対象ベンダーのアカウンティング属性が含まれています。

［Attributes］セクション内で属性の順序を変更することにより、アカウンティング ログ ファイル内の列の順序を変更できます。実際の課金システムで必要としない属性や、使用している機器に適用しない属性がある場合は、その属性を削除するか、コメント アウトできます。これにより、アカウンティング ログ ファイルから作成するスプレッドシートの内容と列順序をデザインできます。

構文は次のとおりです。

```
[Attributes]
属性名 =
属性名 =
```

次に例を挙げます。

```
[Attributes]
User-Name=
NAS-Port=
Framed-IP-Address=
Acct-Status-Type=
Acct-Delay-Time=
Acct-Session-Id=
```

［Attributes］セクションには、属性名のリストを記述します。1 行に 1 つの属性名を記述します。各属性名の直後に必ず等号（=）を入力します。属性名と等号の間にスペースは入れないでください。不正な書式のエントリーは、無効なエントリーとみなされ無視されます。

［Attributes］セクションの各属性名は、RSA RADIUS サーバ上の標準の Dictionary ファイルまたはベンダー固有の Dictionary ファイルで定義されている必要があります。

注：各ログ ファイル エントリーの最初の 6 個の属性（Date、Time、RAS-Client、Record-Type、Full-Name、Auth-Type）は常に有効で、順序を変更したり削除したりすることはできません。したがって、これらの属性は、**account.ini** ファイルの［Attributes］セクションには含まれません。

［Settings］セクション

RSA RADIUS は、現在のアカウンティング ログ ファイル（.act）が閉じられるまでの間、すべてのアカウンティング データを現在のアカウンティング ログ ファイルに書き込みます。このファイルが閉じられると、新しいファイルが開かれ、そのファイルに対してアカウンティング データの書き込みが開始されます。アカウンティング ログ ファイルがこのようにロール オーバーされる頻度は構成可能です。

アカウンティング ログ ファイルの命名規則は、1 日のうちに複数のファイルを生成することが可能なものになっています。次の表は、ロール オーバー期間の違いに応じたファイル命名規則を示しています。以下の例では、*y* は年の桁、*M* は月の桁、*d* は日の桁、*h* は時間の桁、*m* は分の桁を表しています。1 日に複数のファイルを生成する場合は、毎日、_00000 から始まるシーケンス番号 *_nnnnn* が名前に追加されます。

ファイルの生成方法	ファイル命名規則
デフォルト（24 時間）	*yyyyMMdd*.act
非 24 時間ロール オーバー	*yyyyMMdd_hhmm*.act

ファイルの生成方法	ファイル命名規則
サイズによるロール オーバー	*yyyyMMdd_nnnnn*.act
非 24 時間ロール オーバーの場合に、サイズまたは起動によるロール オーバー	*yyyyMMdd_hhmm_nnnnn*.act

account.ini ファイルの［Settings］セクションでは、アカウンティング ログ ファイルにエントリーをどのように書き込むかを制御し、これらのエントリーと各種データベース システム間の互換性を確保します。

次の表は、［Settings］のパラメータとそれらの機能を示しています。

パラメータ	機能
BufferSize	アカウンティング ログ処理で使用するバッファのサイズ（バイト単位） デフォルト値は 131072 バイト。
Carryover	• 1 に設定した場合は、新しいアカウンティング ログ ファイルが作成されるたびに、現在アクティブな各セッションの開始レコードがファイルに書き込まれる。 • 0 に設定した場合は、リストが書き込まれない。 デフォルト値は 1。
Enable	• 1 に設定すると、アカウンティング ログ機能が有効になる。 • 0 に設定した場合は、このサーバ上で .act ファイルが作成されない。 アカウンティング サーバでは、Enable を 1 に設定する必要がある。アカウンティング サーバ以外のサーバでは、処理効率が損なわれないように Enable を 0 に設定すること。 デフォルト値は 1。
LineSize	アカウンティング ログの 1 行の最大サイズを指定する 1,024 ～ 32,768 の範囲内の数値。 デフォルト値は 4,096。

パラメータ	機能
LogFilePermissions	アカウンティング ログ ファイルのオーナーとパーミッションを指定する。 LogFilePermissions の値は、オーナー : グループ パーミッションの形式で入力する。 • オーナー：ファイルの所有者をテキスト形式または数値形式で指定する。 • グループ：ファイルのグループ設定をテキスト形式または数値形式で指定する。 • パーミッション：オーナー / グループ / その他のユーザーがファイルに対してどの権限を行使できるかをテキスト形式または数値形式で指定する。 たとえば、ralphw:1007 rw-r----- と指定した場合は、ファイル所有者（ralphw）にログ ファイルの読み取りと編集の権限が与えられ、グループ 1007 のメンバーにログ ファイルの読み取りの権限が与えられる（編集の権限はなし）。ただし、その他のユーザーはログ ファイルにアクセスできない。
MaxSize	アカウンティング ログ ファイルの最大サイズ（バイト単位）。 アカウンティング ログ ファイルがチェックされた時点で、このサイズに達しているか超えていれば、そのログ ファイルは閉じられ、新しいファイルへの書き込みを開始する。値として 0（デフォルト）を指定すると、サイズの制限がなくなる。 **注：**ログ ファイルのサイズは 1 分おきにチェックされるため、このパラメータで指定した最大サイズよりもログ ファイルが大きくなることがあります。
QuoteBinary	• 1 に設定すると、アカウンティング ログ ファイルに書き込まれたバイナリ値が引用符で囲まれる。 • 0 に設定した場合、引用符は使用されない。 エントリーを処理するアカウンティング アプリケーションで要求される形式に応じて、この値を設定すること。 デフォルト値は 1。
QuoteInteger	• 1 に設定すると、アカウンティング ログ ファイルに書き込まれた整数値が引用符で囲まれる。 • 0 に設定した場合、引用符は使用されない。 エントリーを処理するアカウンティング アプリケーションで要求される形式に応じて、この値を設定すること。 デフォルト値は 1。

パラメータ	機能
QuoteIPAddress	• 1 に設定すると、アカウンティング ログ ファイルに書き込まれた IP アドレスが引用符で囲まれる。 • 0 に設定した場合、引用符は使用されない。 エントリーを処理するアカウンティング アプリケーションで要求される形式に応じて、この値を設定すること。 デフォルト値は 1。
QuoteText	• 1 に設定すると、アカウンティング ログ ファイルに書き込まれたテキスト文字列が引用符で囲まれる。 • 0 に設定した場合、引用符は使用されない。 エントリーを処理するアカウンティング アプリケーションで要求される形式に応じて、この値を設定すること。 デフォルト値は 1。
QuoteTime	• 1 に設定すると、アカウンティング ログ ファイルに書き込まれた日付と時刻の値が引用符で囲まれる。 • 0 に設定した場合、引用符は使用されない。 エントリーを処理するアカウンティング アプリケーションで要求される形式に応じて、この値を設定すること。 デフォルト値は 1。
RollOver	現在のアカウンティング ログ ファイルを閉じて新しいファイルを開く（ロール オーバー）頻度を指定する。ロール オーバーの頻度は、最大で 1 分に 1 回まで増やすことができる。ゼロ以外の値は、次回のロール オーバーを行うまでの時間（分単位）を示す。 0 に設定すると、アカウンティング ログ ファイルが 24 時間に 1 回（ローカル時間で深夜 0 時に）ロール オーバーされる。 デフォルト値は 0。
RollOverOnStartup	• 1 に設定した場合は、RSA RADIUS が起動されるたびに、現在のアカウンティング ログ ファイルが閉じられ、新しいファイルが開かれる。MaxSize に達した場合と同様に、シーケンス番号 *_nnnnn* がログ ファイル名に付加される。 • 0 に設定した場合は、RSA RADIUS が起動されるたびに、前回開かれていたアカウンティング ログ ファイルにエントリーが追加される。 デフォルト値は 0。

パラメータ	機能
Titles	• 1 に設定した場合は、新しいアカウンティング ログ ファイルが作成されるたびに、タイトル行（列見出しを含む行）がファイルに書き込まれる。 • 0 に設定した場合、タイトル行は書き込まれない。 デフォルト値は 1。
UTC	• 1 に設定した場合は、協定世界時（Universal Time Coordinated：UTC、旧称はグリニッジ標準時（GMT））に基づく時刻と日付の値が使用される。 • 0 に設定した場合は、ローカル時間の時刻と日付の値が使用される。 デフォルト値は 0。

［TypeNames］セクション

account.ini の［TypeNames］セクションの各エントリーにより、Acct-Status-Type 属性に設定される値を文字列にマッピングできます。この属性の値は、各アカウンティング ログ レコードの 4 番目の列に書き込まれます。

構文は次のとおりです。

```
[TypeNames]
タイプ ID = タイプ名
タイプ ID = タイプ名
```

次の表は、［TypeNames］のパラメータとそれらの機能を示しています。

パラメータ	機能
タイプ ID	各タイプ *ID* には、Acct-Status-Type 属性に設定される可能性のある数値を指定する。この属性は、すべての着信 RADIUS アカウンティング パケットに含まれており、パケットに格納されている可能性の高いデータのタイプを識別する。
タイプ名	各タイプ名を文字列で指定する。この文字列は、パケットのタイプを識別するためにアカウンティング ログに書き込まれる。

標準の Acct-Status-Type の値である 1、2、3、7、8 は、次に示すように、**account.ini** の［TypeNames］セクションにすでに追加されています。

```
[TypeNames]
1=Start
2=Stop
3=Interim
7=On
8=Off
```

［TypeNames］セクションを編集して、ベンダー固有のパケット タイプをこのリストに追加できます。これにより、アカウンティング ログ ファイルをより読みやすく、また使いやすくすることができます。次に例を挙げます。

```
[TypeNames]
1=Start
2=Stop
3=Interim
7=On
8=Off
639=AscendType
28=3ComType
```

特定の Acct-Status-Type に対して文字列が指定されていない場合、RSA RADIUS は着信 Acct-Status-Type 属性の数値を、文字列形式で記録します。

6 EAP 構成ファイル

この章では、自動 EAP ヘルパ メソッドのオプションを指定する EAP 構成ファイルおよびヘルパ ファイルについて説明します。これらのファイルは RSA RADIUS ディレクトリ内に存在し、起動時にロードされます。

注：構成ファイルおよびヘルパ ファイルには、現在のリリースの RSA RADIUS では使用しないパラメータやセクションが含まれていることがあります。本書では、現在のリリースに関連しているセクションおよびパラメータについてのみ説明しています。本書で説明していないパラメータは変更しないでください。

eap.ini ファイル

注：eap.ini ファイルの設定には RSA Operations Console を使用します。**eap.ini** ファイルは手動で編集しないでください。

eap.ini 構成ファイルは、さまざまな RSA RADIUS 認証方式でユーザーを認証する場合に、試行する EAP 認証タイプの順番を構成します。

EAP 認証を実行する各認証方式が、この **eap.ini** ファイル内で構成されている必要があります。

このファイルでは、使用する認証方式ごとにセクションを 1 つ構成する必要があります。また、各セクションには、認証方式を識別できるタイトルを付ける必要があります。

- SecurID
- EAP-TTLS
- EAP-PEAP

次の表は、各セクションで使用するパラメータを示しています。

パラメータ	機能
EAP-Only	• この値を 0 に設定すると、認証方式はすべてのタイプのユーザー認証情報を受け付ける。 • この値を 1 に設定すると、認証方式には EAP 認証情報のみが渡されるか、または自動 EAP プロトコル メソッドのバックエンド サーバとしてのみ機能する。 EAP-TTLS 内部認証に対応する認証方式の場合は、内部認証に使用する認証情報のタイプに基づいて、このパラメータを 0 または 1 のどちらに設定するかを決定する必要がある。 **注：**PEAP で SecurID を使用する場合は、このパラメータを 0 に設定してください。PEAP プラグ インではセキュリティ上の理由により内部 EAP/Generic Token 認証情報が PAP に変換されるため、このパラメータを 1 に設定すると、EAP/Generic Token の使用時に SecurID の処理がスキップされ、ユーザーが拒否される結果につながります。
EAP-Type	この認証方式をサポートする EAP プロトコルのカンマ区切りリスト。リストの先頭のプロトコルがプライマリ プロトコルとなる。リスト内の 2 番目以降のプロトコルがこの認証方式とともに使用されるのは、クライアントが EAP NAK を返して応答し、そのプロトコルを指定した場合、または他の認証方式がそのプロトコルの使用をトリガーしたが要求を完了できなかった場合のみである。 有効な値は次のとおり。 • SecurID – EAP-15 – EAP-32 – Generic-Token • EAP-TTLS – TTLS • EAP-PEAP – PEAP 認証方式に対して EAP を無効にするには、EAP-Type リストに何も指定しない。

パラメータ	機能
First-Handle-Via-Auto-EAP	• この値を 1 に設定すると、ユーザー認証情報が EAP である場合に、認証方式の前に適切な自動 EAP ヘルパ メソッドが呼び出される。自動 EAP ヘルパ メソッドは、ユーザーの EAP 認証情報を、認証方式がサポートしている形式に変換するために呼び出される。 • この値を 0 に設定すると、自動ヘルパ メソッドが呼び出される前に、認証方式自体が要求を直接処理する。 デフォルトは、ユーザーのタイプによって異なる。詳細については、**eap.ini** ファイル内のコメントを参照。 **注：**マシン認証を使用する場合は、**eap.ini** ファイルの［Windows Domain User］セクションおよび［Windows Domain Group］セクションで、このパラメータの値として **1** を指定する必要があります。 **注：**マシン認証を使用する場合は、**winauth.aut** ファイルの［WindowsDomain］セクションで AllowMachineLogin パラメータを **Yes** に設定する必要があります。
Available-EAP-Types	セキュリティ コンソールを通じて RSA RADIUS サーバを構成するときに選択できる EAP プロトコルのカンマ区切りリスト。 有効な値は次のとおり。 • TTLS • Generic-Token

peapauth.aut ファイル

注：peapauth.aut ファイル内の設定を編集する場合は、標準的な .ini 構文に準拠する必要があります。**peapauth.aut** ファイルに何らかの変更を加えた後は、それらの変更を反映するために RADIUS サーバを再起動する必要があります。

EAP-PEAP プラグ インの設定は、**peapauth.aut** ファイルに格納されます。**peapauth.aut** 構成ファイルは、RSA RADIUS サーバを再起動するたびに読み込まれます。

［Bootstrap］セクション

peatauth.aut ファイルの［Bootstrap］セクションでは、RSA RADIUS が EAP-PEAP 認証方式のロードに使用する情報を指定します。

次の表は、［Bootstrap］のパラメータとそれらの機能を示しています。

パラメータ	機能
LibraryName	EAP-PEAP モジュールの名前を指定する。デフォルト値は **peapauth.so**。RSA カスタマー サポートから指示があった場合以外は、この値を変更しないこと。
Enable	EAP-PEAP 認証モジュールを有効にするかどうかを指定する。 • この値を 0 に設定すると、EAP-PEAP が無効になり、［Authentication Policies（認証ポリシー）］パネルの［Authentication Methods（認証方式）］リストに EAP-PEAP 認証方式が表示されなくなる。 • この値を 1 に設定すると、EAP-PEAP が有効になる。 デフォルト値は 0。
InitializationString	［Authentication Policies（認証ポリシー）］パネルの［Authentication Methods（認証方式）］リストに表示する認証方式の名前を指定する。 各認証方式の名前は一意でなければならない。複数のデータベースに対して認証を実装するために追加の .aut ファイルを作成する場合は、各ファイルの InitializationString パラメータの値に一意の方式名を指定する必要がある。 デフォルト値は EAP-PEAP。

［Server_Settings］セクション

［Server_Settings］セクションでは、EAP-PEAP プラグ インの基本動作を構成します。

次の表は、［Server_Settings］のパラメータとそれらの機能を示しています。

パラメータ	機能
TLS_Message_Fragment_Length	TLS 交換が繰り返されるたびに生成される可能性がある TLS メッセージの最大サイズを設定する。 アクセス ポイントによっては、RADIUS 応答または EAP メッセージが Ethernet フレーム 1 つのサイズ（IP/UDP ヘッダを含む 1500 バイト）を超えている場合に問題が発生する可能性がある。 デフォルト値（1020）を使用すると、RADIUS チャレンジ応答（UDP パケットで伝送）のサイズが Ethernet フレーム 1 つのサイズを超えることを防止できる。通常は、これが最も安全な設定となる。 値を小さく設定すると、TLS 交換を完了するまでに必要となる RADIUS チャレンジ / 応答の往復回数に影響が生じる。値を 1400 に設定すると往復が 6 回、500 に設定すると往復が 15 回になる可能性がある。 最小値は 500。
Return_MPPE_Keys	この属性を 1 に設定すると、モジュールはアクセス ポイントに送信する最後の RADIUS Accept 応答に RADIUS MS-MPPE-Send-Key 属性および MS-MPPE-Recv-Key 属性を含める。これは、アクセス ポイントが WEP 暗号化キーを生成するために必要となる。 アクセス ポイントで認証される対象がエンド ユーザーのみであり、WEP が使用されない場合は、この属性を 0 に設定することができる。 デフォルト値は 1。
DH_Prime_Bits	モジュールが Diffie-Hellman 累乗法に使用する素数のサイズを指定する。大きい素数を選択すると、システムが特定のタイプの攻撃に影響されにくくなるが、Diffie-Hellman 鍵交換の演算に伴う CPU 負荷が高くなる。 有効値は、512、1024、1536、2048、3072、4096。 デフォルト値は 1024。

パラメータ	機能
Cipher_Suites	サーバが使用する TLS 暗号スイートのリストを（優先度順に）指定する。これらの暗号スイートは、RFC 2246「TLS プロトコル バージョン 1」や、TLS 関連のその他の RFC およびドラフト RFC に記述されている。 デフォルト値は、0x16、0x13、0x66、0x15、0x12、0x0a、0x05、0x04、0x07、0x09。
PEAP_Min_Version	サーバがネゴシエートする PEAP プロトコルの最小バージョンを指定する。 • この値を 0 に設定すると、サーバはバージョン 0 をネゴシエートする。バージョン 0 は、Microsoft による初期の PEAP 実装（Microsoft XP Service Pack 1 に搭載された実装版）と互換性がある。 • この値を 1 に設定すると、サーバはバージョン 1 をネゴシエートする。バージョン 1 は、Cisco による初期の PEAP 実装（Cisco ACU に搭載された実装版）と互換性がある。 デフォルト値は 0。 **注：**このパラメータに入力した値は、PEAP_Max_Version パラメータに入力した値以下になっている必要があります。
PEAP_Max_Version	サーバがネゴシエートする PEAP プロトコルの最大バージョンを指定する。 • この値を 0 に設定すると、サーバはバージョン 0 をネゴシエートする。バージョン 0 は、Microsoft による初期の PEAP 実装（Microsoft XP Service Pack 1 に搭載された実装版）と互換性がある。 • この値を 1 に設定すると、サーバはバージョン 1 をネゴシエートする。バージョン 1 は、Cisco による初期の PEAP 実装（Cisco ACU に搭載された実装版）と互換性がある。 デフォルト値は 1。 **注：**このパラメータに入力した値は、PEAP_Min_Version パラメータに入力した値以上になっている必要があります。

［Session_Resumption］セクション

［Session_Resumption］セクションでは、セッション再利用を許可するかどうか、許可する場合はどのような条件下でセッション再利用を実行するかを指定します。

注：セッション再利用を行うには、Session-Timeout リターンリスト属性に対応するようネットワーク アクセス デバイスを構成する必要があります。これは、タイマーがタイムアウトになった後に、ネットワーク アクセス デバイスがクライアントに再認証を指示できるようにする必要があるためです。

次の表は、［Session_Resumption］のパラメータとそれらの機能を示しています。

パラメータ	機能
Session_Timeout	この属性では、再認証が必要となるまでの、クライアントをネットワーク アクセス デバイスに接続した状態に維持しておく最長時間を秒単位で設定する。 • 0 より大きい値に設定すると、この値と残りのセッション再利用制限（この後の説明を参照）のうち、どちらか小さい方が RADIUS Access Accept 応答の Session-Limit 属性に格納されて、RADIUS クライアントに送信される。 • この値を 0 に設定すると、Session-Limit 属性がプラグ イン によって生成されない。ただし、二次承認を実行する認証方式がこの属性の値を設定することを妨げない。 デフォルト値は 0。 600（10 分）などの値を入力しても、再認証のために、10 分おきに完全な認証が必要になるとは限らない。セッション再利用を構成することにより、ほとんどの再認証処理を高速化し、CPU への負荷を抑えることが可能。
Termination_Action	この属性では、Termination-Action 属性で返す整数値を設定する。これは、ほとんどのアクセス ポイントでサポートされる標準的な属性であり、セッション タイムアウトに達したときの処理を決定する。 この属性の値を省略した場合、プラグ インはこれに該当する属性を生成しない。ただし、二次承認を実行する認証方式がこの属性の値を設定することを妨げない。 デフォルトでは、この属性は送信されない。

パラメータ	機能
Resumption_Limit	この属性では、クライアントが TLS セッション再利用機能を使用して再認証を受けることが可能な最長時間を秒単位で設定する。 このタイプの再認証は、高速で、CPU 負荷が低い。ただし、このタイプの再認証は以前の認証に依存するため、完全な（CPU への負荷が高い）認証に比べて安全性の点で劣るとみなされる場合がある。この値を 0 に設定すると、セッション再利用機能が無効になる。 デフォルト値は 0。

ttlsauth.aut ファイル

注：ttlsauth.aut ファイル内の設定を編集する場合は、標準的な .ini 構文に準拠する必要があります。**ttlsauth.aut** ファイルに何らかの変更を加えた後は、それらの変更を反映するために RADIUS サーバを再起動する必要があります。

EAP-TTLS 認証方式の設定は、**ttlsauth.aut** ファイルに格納されます。**ttlsauth.aut** 構成ファイルは、RSA RADIUS サーバを再起動するたびに読み込まれます。

［Bootstrap］セクション

ttlsauth.aut ファイルの［Bootstrap］セクションでは、RSA RADIUS が EAP-TTLS 認証方式のロードに使用する情報を指定します。

次の表は、［Bootstrap］のパラメータとそれらの機能を示しています。

パラメータ	機能
LibraryName	EAP-TTLS モジュールの名前を指定する。デフォルト値は ttlsauth.so。RSA カスタマー サポートから指示があった場合以外は、この値を変更しないこと。
Enable	EAP-TTLS 認証モジュールを有効にするかどうかを指定する。 • この値を 0 に設定すると、EAP-TTLS が無効になり、［Authentication Policies（認証ポリシー）］パネルの［Authentication Methods（認証方式）］リストに EAP-TTLS 認証方式が表示されなくなる。 • この値を 1 に設定すると、EAP-TTLS が有効になる。 デフォルト値は 0。

パラメータ	機能
InitializationString	［Authentication Policies（認証ポリシー）］パネルの［Authentication Methods（認証方式）］リストに表示する認証方式の名前を指定する。 各認証方式の名前は一意でなければならない。複数のデータベースに対して認証を実装するために追加の .aut ファイルを作成する場合は、各ファイルの InitializationString パラメータの値に一意の方式名を指定する必要がある。 デフォルト値は EAP-TTLS。

［Server_Settings］セクション

［Server_Settings］セクションでは、EAP-TTLS プラグ インの基本動作を構成します。

次の表は、［Server_Settings］のパラメータとそれらの機能を示しています。

パラメータ	機能
TLS_Message_Fragment_Length	TLS 交換が繰り返されるたびに生成される可能性がある TLS メッセージの最大サイズを設定する。この値は、TLS 交換を完了するまでに必要となる RADIUS チャレンジ / 応答の往復回数に影響を及ぼす。値を 1400 に設定すると往復が 6 回、500 に設定すると往復が 15 回になる可能性がある。 アクセス ポイントによっては、RADIUS 応答または EAP メッセージが Ethernet フレーム 1 つのサイズ（IP/UDP ヘッダを含む 1500 バイト）を超えている場合に問題が発生する可能性がある。 最小値は 500。 最大値は 4096。 デフォルト値は 1020。この値を使用すると、RADIUS チャレンジ応答（UDP パケットで伝送）のサイズが Ethernet フレーム 1 つのサイズを超えることを防止できる。

パラメータ	機能
Return_MPPE_Keys	この属性を 1 に設定すると、モジュールはアクセス ポイントに送信する最後の RADIUS Accept 応答に RADIUS MS-MPPE-Send-Key 属性および MS-MPPE-Recv-Key 属性を含める。これは、アクセス ポイントが WEP 暗号化キーを生成するために必要となる。 アクセス ポイントで認証される対象がエンド ユーザーのみであり、WEP が使用されない場合は、この属性を 0 に設定することができる。 デフォルト値は 1。
DH_Prime_Bits	モジュールが Diffie-Hellman 累乗法に使用する素数のサイズを指定する。大きい素数を選択すると、システムが特定のタイプの攻撃に影響されにくくなるが、Diffie-Hellman 鍵交換の演算に伴う CPU 負荷が高くなる。 有効値は、512、1024、1536、2048、3072、4096。 デフォルト値は 1024。
Cipher_Suites	サーバが使用する TLS 暗号スイートのリストを（優先度順に）指定する。これらの暗号スイートは、RFC 2246「TLS プロトコル バージョン 1」や、TLS 関連のその他の RFC およびドラフト RFC に記述されている。 デフォルト値は、0x16、0x13、0x66、0x15、0x12、0x0a、0x05、0x04、0x07、0x09。
Require_Client_Certificate	• この値を 1 に設定すると、クライアントが TTLS 交換中に証明書を提示することが必須になる。 • この値を 0 に設定した場合、クライアント証明書は要求されない。 デフォルト値は 0。

［Session_Resumption］セクション

［Session_Resumption］セクションでは、セッション再利用を許可するかどうか、許可する場合はどのような条件下でセッション再利用を実行するかを指定します。

注：セッション再利用を行うには、Session-Timeout リターンリスト属性に対応するようネットワーク アクセス デバイスを構成する必要があります。これは、タイマーがタイムアウトになった後に、ネットワーク アクセス デバイスがクライアントに再認証を指示できるようにする必要があるためです。

次の表は、［Session_Resumption］のパラメータとそれらの機能を示しています。

パラメータ	機能
Session_Timeout	この属性では、再認証が必要となるまでの、クライアントをネットワーク アクセス デバイスに接続した状態に維持しておく最長時間を秒単位で設定する。 • 0 より大きい値に設定すると、この値と残りのセッション再利用制限（この後の説明を参照）のうち、どちらか小さい方が RADIUS Access Accept 応答の Session-Limit 属性に格納されて、ネットワーク アクセス デバイスに送信される。 • この値を 0 に設定すると、Session-Limit 属性がプラグ イン によって生成されない。ただし、二次承認を実行する認証方式がこの属性の値を設定することを妨げない。 デフォルト値は 0。 600（10 分）などの値を入力しても、再認証のために、10 分おきに完全な認証が必要になるとは限らない。セッション再利用を構成することにより、ほとんどの再認証処理を高速化し、CPU への負荷を抑えることが可能。
Termination_Action	この属性では、Termination-Action 属性で返す整数値を設定する。これは、ほとんどのアクセス ポイントでサポートされる標準的な属性であり、セッション タイムアウトに達したときの処理を決定する。 この属性の値を省略した場合、プラグ インはこれに該当する属性を生成しない。ただし、二次承認を実行する認証方式がこの属性の値を設定することを妨げない。 デフォルトでは、この属性は送信されない。
Resumption_Limit	この属性では、クライアントが TLS セッション再利用機能を使用して再認証を受けることが可能な最長時間を秒単位で設定する。 このタイプの再認証は、高速で、CPU 負荷が低い。ただし、このタイプの再認証は以前の認証に依存するため、完全な（CPU への負荷が高い）認証に比べて安全性の点で劣るとみなされる場合がある。この値を 0 に設定すると、セッション再利用機能が無効になる。 デフォルト値は 0。

［Integrity_Settings］セクション

［Integrity_Settings］セクションでは、隔離対象として指定されたユーザーの処理方法をオプションの Endpoint Assurance Server ソフトウェアで指定する際に使用可能な、隔離プロファイルのリストを指定します。

```
[Integrity_Settings]
;Quarantine_Profiles=QUARANTINE QUARANTINE2
```

次の表は、[Integrity_Settings] のパラメータとその機能を示しています。

パラメータ	機能
Quarantine_Profiles	隔離対象として指定されたユーザーに対して Endpoint Assurance Server ソフトウェアで割り当てることが可能な RSA RADIUS プロファイルのリストを指定する。 複数のプロファイル名を入力するには、すべての名前を同じ行に入力し、各プロファイル名を空白で区切る。 デフォルトでは、隔離プロファイルはない。

ttlsauth.aut ファイルのサンプル

```
[Bootstrap]
LibraryName=ttlsauth.dll
Enable=1
InitializationString=EAP-TTLS

; Maximum TLS Message fragment length EAP-TLS will handle.
TLS_Message_Fragment_Length = 1020

; Indicates whether the EAP-TLS module should return the
; MS-MPPE-Send-Key and MS-MPPE-Recv-Key attribute upon
successful
; authentication of user.
Return_MPPE_Keys = 1

; Size of the prime to use for DH modular exponentiation.
DH_Prime_Bits = 1536
; TLS cipher suites (in order of preference)
; that the server is to use.
Cipher_Suites = 0x16, 0x13, 0x66, 0x15, 0x12, 0x0a, 0x05,
0x04, 0x07, 0x09
```

索引