

WebLine Media Blender™ Configuration Handbook



Version 3.0

Contents

In this guide:

[Section 1: Media Blender Overview](#)

Read this portion of the guide to obtain a general understanding of Media Blender and how it fits into a call center environment. Section 1 contains these sections:

[About Media Blender](#)

[Understanding Blended Collaboration](#)

[Media Blender Configuration Checklist](#)

[Section 2: Basic Configuration](#)

Use this portion of the guide in conjunction with your *Media Blender Installation Guide* when setting up Media Blender for the first time. Section 2 contains information about the minimum set up required to get Media Blender up and running. It contains these sections:

[About configuration files](#)

[Identify participating media](#)

[Configure the connection to Media Blender](#)

- [Collaboration medium](#)
- [Aspect medium](#)
- [ASAI medium](#)
- [CT-Connect medium](#)
- [JTAPI medium](#)
- [Lucent ICM medium](#)
- [Media Blender Telephony Integration API](#)

[Establish call types and outbound dialing strategies](#)

- [Strategies for different call types](#)
- [Mapping call types to CTI strategies](#)
- [Identifying the calltype file to your ACD medium](#)

- [Define a Pool of Phantom Lines](#)
- [If you are using only one calltype](#)

[Set up agent log in](#)

- [Unblended log in](#)
- [Blended log in: If agents always use the same phone](#)
- [Blended log in: If agents frequently use different phones](#)

[Set Up Skills-Based Routing](#)

- [Map Web-based skills to ACD logic](#)
- [Identify your skill table to your ACD medium](#)

[Driver-specific setup tasks](#)

- [Aspect Media: Configure Eventbridge](#)
- [CT-Connect driver](#)
 - [Determine CT-Connect Monitor Requirements](#)
 - [Install the CT-Connect API](#)
 - [Create an Agent Group File \(CSTA switches only\)](#)
 - [Create a Position ID file \(Nortel switches only\)](#)
- [Lucent ICM driver](#)

[Section 3: Advanced/Optional Configuration](#)

Use this section after you've ensured the basic installation and configuration of Media Blender is sound. Section 3 contains these sections:

[Setting up your log file](#)

[Configure the Collaboration Server](#)

- [Set up the Blender Agent](#)
- [Tailor your callBack page](#)
- [Turn off meetMe AutoConnect](#)

[Set up email notification](#)

[Using Multiple Media Blenders](#)

- [Create an agent filter file](#)

[Appendix A: Terminology](#)

Refer to Appendix B for definitions of Media Blender and other WebLine terminology.

[Copyright](#)

Additional Information:

Media Blender Reference Guide

Media Blender Switch Administrator's Guide: Aspect Callcenter

Media Blender Switch Administrators Guide: Lucent Definity Generic 3

Section 1: Media Blender Overview

Before you begin, you should have a good understanding of Media Blender and the blended Collaboration environment. This section describes the function of Media Blender, how it fits within your configuration, and how it affects WebLine Collaboration. This section also includes a checklist you can use when setting up Media Blender for the first time.

In this section:

- [About Media Blender](#)
- [Understanding Blended Collaboration](#)
- [Media Blender Configuration Checklist](#)

About Media Blender

Media Blender allows you to synchronize your Web-based and ACD-based call center systems by sharing CTI events among participating media. It serves as a backbone for blending call requests and calls from disparate sources (such as the World Wide Web) in a high-volume call center environment. The typical scenario for this integration is in joining the Web-based visual service of WebLine Collaboration Server with the call processing power of an ACD.

Components of a *blended* configuration

Media Blender shares events between participating *media*, or electronic forms of session-based customer interaction, such as an ACD or a Web-based application. A typical blended configuration shares CTI events between these media:

ACD medium

The ACD medium is Java-based software package designed to handle CTI messages coming from a high-density ACD, Predictive Dialer, or PBX. The same ACD medium is used for each telephony implementation, with a specific CTI Driver written for each system. CTI drivers are configurable. Media Blender supports these CTI packages:

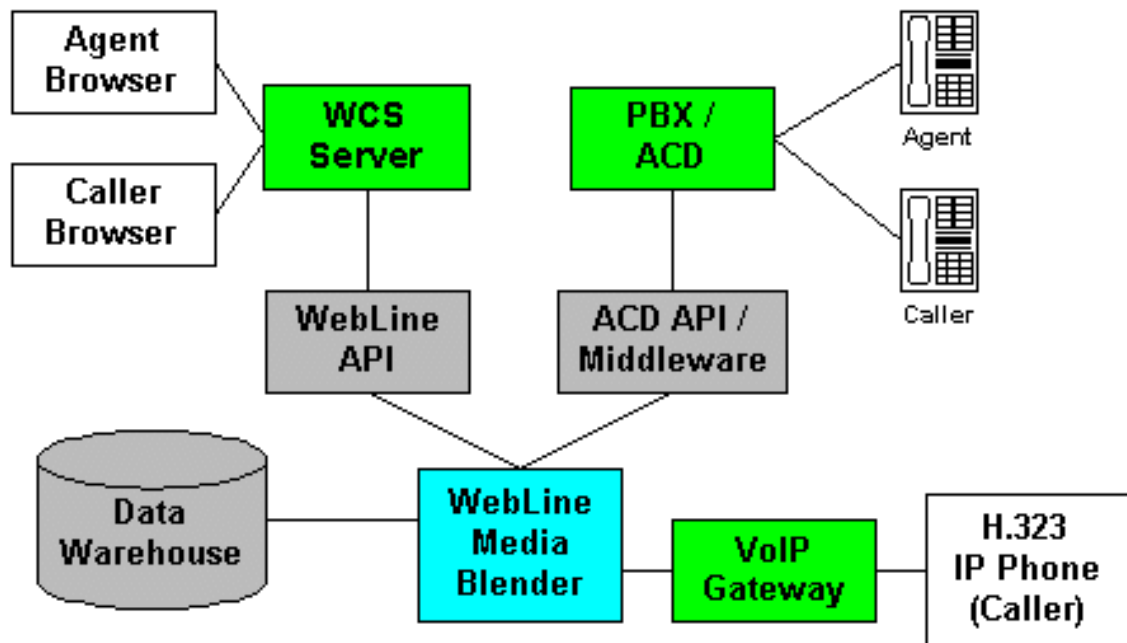
- ASAI
- Aspect Application Bridge (version 6 and above)
- Lucent Internet Call Center
- Java Telephony API (JTAPI)
- Dialogic CT-Connect
- Media Blender Telephony Integration API

Web medium

Media Blender also shares CTI events with a *Web medium*, specifically, the WebLine Collaboration Server (WCS) medium. The WCS medium is a Java-based software package that receives notifications of Collaboration sessions and to control the sessions.

Media Blender configuration also includes Web-based administration tools. Media Blender Administrator can troubleshoot potential problems using only a Web browser.

An architectural view of Media Blender and its media is illustrated below:



Event sharing

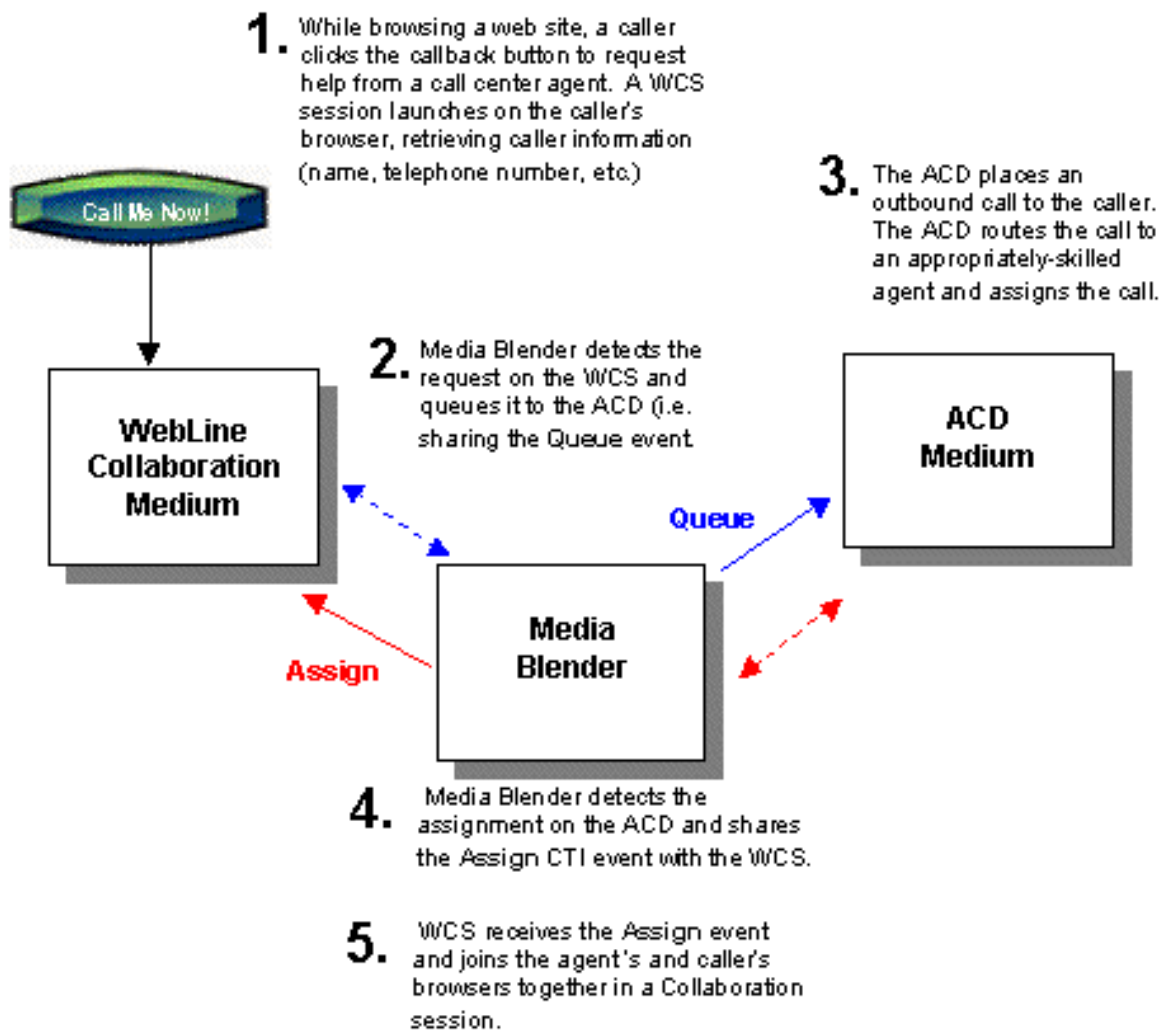
Media Blender acts as an event bus sharing software events between the the WebLine Collaboration Server and the ACD already implemented at your site. For instance

The shared events are session-based information, such as:

- Agent signon
- Session (or call) queue from a caller
- Session (or call) assignment to an agent
- Session (or call) transfer to another agent
- Session (or call) conference with another agent or caller
- Session (or call) termination
- Session (or call) progress updates
- Agent status change
- Agent signoff

When a medium notices that one of these events has taken place internally, for example when an ACD Medium receives CTI notification of a queued call, it packages this event into a unified Blender format. This event is given to Media Blender, which then shares the event with all other media. Each medium can be configured to filter which events it shares or accepts, from specific other media. The net effect of this *Event Bus* architecture is that all relevant calls and sessions are blended across all media.

The following example illustrates how Media Blender shares Queue and Assign events between the WCS and the ACD medium:



Understanding calltypes and outbound callflow

You can configure Media Blender so that your configuration can provide callback for a number of different calltypes; that is, with Media Blender, you can respond to callers via voice, chat, and voice-over-IP (VoIP) call types.

Media Blender can then use a number of different outbound dialing strategies, or *CTI strategies*, to determine how your ACD places the outbound call to the caller. based on calltype CTI strategies tell Media Blender how to place the outbound call to the caller.

Predictive CTI Strategy--Predictive CTI Strategy assumes that the ACD or PBX is capable of recognizing how an outgoing call is answered. This strategy requires a predictive dialer, which can detect a busy signal when the call is answered, as well as whether the call is answered by an answering machine rather than an actual caller. Predictive strategy also assumes that the ACD or PBX has the ability to place an outgoing call into an inbound ACD queue.

Phantom Line CTI Strategy--Blender uses a pool of *phantom lines*. For each request, Media Blender generates a call to the ACD from one of the phantom lines from the pool. The phantom line then waits in the ACD queue on behalf of the caller.

A phantom strategy may require that you set aside a number of physical phones to act as phantom callers on behalf of the actual Web requestors. (Some ACDs support virtual phantom phones; see the Media Blender Switch Administrator's guide for information on setting up phantom lines.) When a request arrives, Media Blender uses one of these *phantoms* to dial into the ACD and sit in queue on behalf of the caller. When an agent is available, the phantom is released, and Media Blender causes the agent's phone to dial back to the caller.

There are several variations of phantom line strategies, which accommodate different kinds of callBack:

- PhantomWaitRelease
- PhantomWaitNoRelease
- PhantomNoCallRelease
- PhantomNoCallNoRelease

Von CTI Strategy--

The Von CTI Strategy is used when Media Blender is configured to receive Voice-Over-IP calls via a Voice-Over IP gateway.

Note: Voice-Over-IP supported only under special agreements with WebLine, and is not standard with this release of Media Blender.

See [Establish call types and outbound dialing strategies](#) for more information on how to use CTI strategies.

Understanding skills-based routing

Media Blender can match web-based routing codes (embedded on the callBack form submitted by the caller) with routing addresses set up on your switch. You can therefore take advantage of the skills-based routing capabilities provided by your ACD.

Note that appropriate routing addresses need to be set up on your ACD by your switch administrator. We provide these guides to assist your switch administrator in setting up WCS routing addresses:

- *Media Blender Switch Administrator's Guide: Aspect CallCenter*
- *Media Blender Switch Administrator's Guide: Lucent Definity Generic 3*

Understanding Blended Collaboration

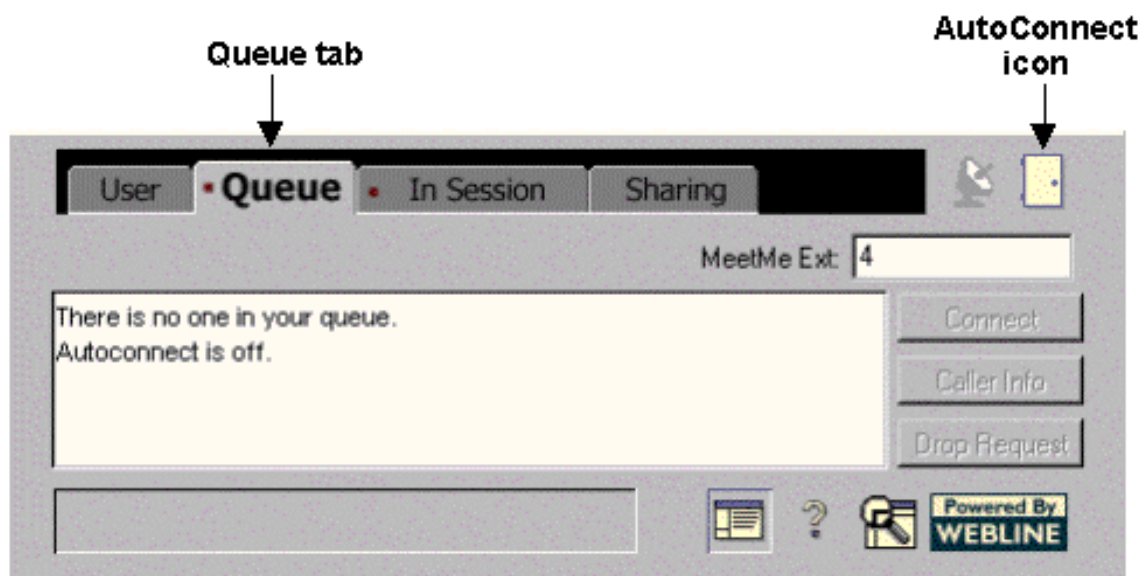
The WCS is designed to handle different types of requests based on a site's configuration. Different types of WCS requests are:

- callBack requests--requests routed to an agent by an ACD using WebLine Media Blender
- meetMe requests--requests initiated by a phone call to a call center agent. The agent provides the caller with a temporary or permanent extension, which the caller then uses to initiate a WCS session

The Agent UI displays various options and icons that apply to each kind of potential request. Although, from an agent's point of view, there is little difference between Blended and unblended Collaboration, both administration and agents in a blended configuration should understand these features of the WCS agent UI:

Queue tab and AutoConnect icon

Agents should be aware of how the Queue tab and the AutoConnect icon functions in WCS and how it relates to blended requests.



Queue tab

The Queue tab on the agent interface shows only meetMe requests; it does not display calls routed to the ACD by Media Blender. Blended Collaboration agents should be aware that calls queued to them will not appear in the Queue tab. The calls appear instead on the ACD phone sets.

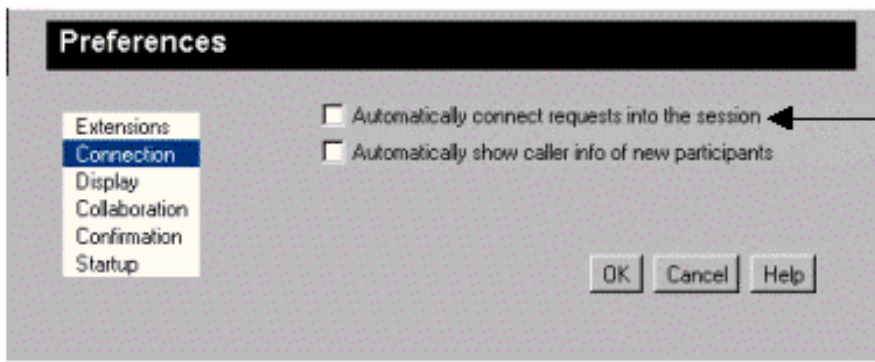
AutoConnect

The AutoConnect icon is used only for meetMe requests and does not affect blended requests. You can ensure this icon does NOT appear on the interface of agents using Blended Collaboration by configuring the agents.properties file on the WCS. See Turn off meetMe AutoConnect in the WCS setup section for information on how to turn AutoConnect off.

Important: Individual agents, however, have the ability to turn AutoConnect on using the AutoConnect user connection preference (see description below). Ensure agents are aware that this feature should remain deactivated.

User connection and startup preferences

Agent's should be aware of how the AutoConnect connection preference function within WCS and how it relates to blended requests:



AutoConnect preference This preference is used for meetMe requests only. Ensure this preference is deselected for agent's using blended Collaboration.

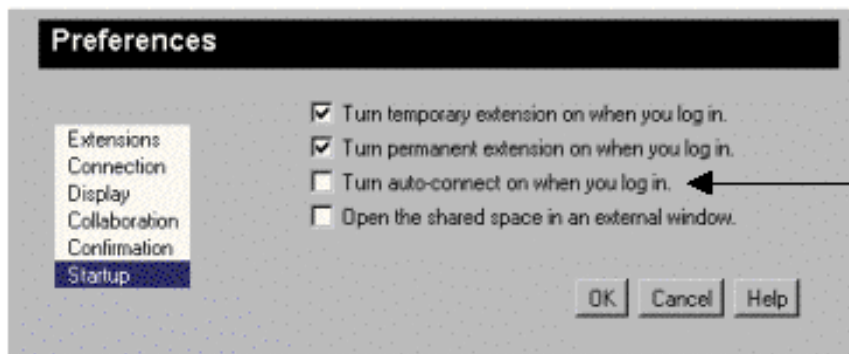
AutoConnect connection preference

This preference turns the AutoConnect feature, which allows meetMe requests to go directly into session with an agent as they arrive. Note that, even if you have removed the AutoConnect icon from the interface, agents can still set this preference. Ensure that agents using blended Collaboration understand that they should keep this preference deselected.

Agents can also specify that the AutoConnect feature should be turned on each time the agent logs in, as explained in the following section:

AutoConnect startup preference

Agent's should be aware of how the AutoConnect startup preference functions within WCS and how it relates to blended requests.



AutoConnect startup preference This preference turns AutoConnect on each time the agent logs in. Agents using blended Collaboration should ensure this preference is deselected.

The AutoConnect startup preferences specifies that AutoConnect should be turned on each time the agent logs in. If it is selected, AutoConnect will be turned on, regardless of whether:

- the AutoConnect icon appears on the agent's UI
- the AutoConnect connection preference is selected or not

Ensure that agents using blended Collaboration understand that they should keep this preference deselected.

For more information on the agent WCS interface, consult these references:

- WCS online help
- *Agent Quick Start*
- *Agent Quick Reference*
- *Cyberseminar Quick Start*
- *Cyberseminar Quick Reference*
- *WebLine Collaboration Administrator's Guide*

Media Blender Configuration Checklist

Use the Basic Configuration checklist below to view the tasks necessary to configure Media Blender for the first time. Also refer to Appendix A for

Use the Advanced/Optional Configuration checklist to view tasks you can set up after basic configuration is stable.

Basic Configuration

- a [Identify participating media](#)
- a [Configure the connection to Media Blender](#)
- a [Establish call types and outbound dialing strategies](#)
- a [Set up agent log in](#)
- a [Setting Up Skills-Based Routing](#)
- a [Driver-specific setup tasks](#)

Advanced/Optional Configuration

- a [Setting up your log file](#)
- a [Configure the Collaboration Server](#)
- a [Set up email notification](#)
- a [Using Multiple Media Blenders](#)

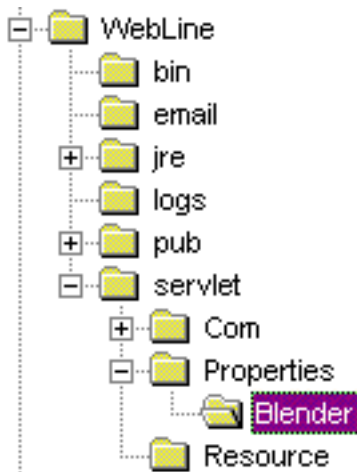
Section 2: Basic Configuration

Once the Media Blender software has been installed, you must configure both the Media Blender and all participating media to ensure proper connectivity and to set up such features as call routing and agent login. In this section:

- [About configuration files](#)
- [Identify participating media](#)
- [Configure the connection to Media Blender](#)
- [Establish call types and outbound dialing strategies](#)
- [Set up agent log in](#)
- [Set Up Skills-Based Routing](#)
- [Driver-specific setup tasks](#)

About configuration files

You configure Media Blender and all participating media using a number of plain text *property* files. These files contain simple name value pairs (properties) that define the behavior of the Media Blender or medium. You can modify property files with any text editor. They typically exist in this directory:



The folder above contains many properties files; not all of these are needed for every installation. The files that are used in every install are:

Property file	Description
blender.properties	Media Blender configuration
ACD.xxx.properties	ACD medium configuration, where xxx represents the medium in use at your installation. For instance, ACD.aspect.properties identifies the property file for an Aspect ACD.
Collaboration.properties	WebLine Collaboration Server configuration.

Other configuration files you are likely to use are:

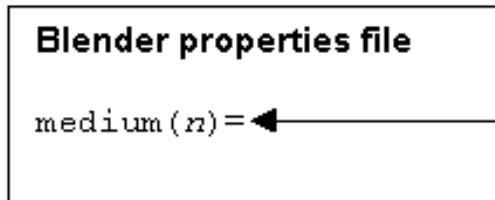
Property file	Description
calltypes.properties	Used to define the outbound callflow for different call types.
agents.properties passwords.properties	Used for mapping agent's web-based log in information with physical phone IDs and passwords.
skills.properties	Used for mapping web-based routing codes to existing ACD routing logic

Identify participating media

You configure the behavior of Media Blender in this properties file: <weblinedir>\servlet\properties\blender.properties. You use this file to configure and improve the performance of Media Blender and to define all media in your configuration to Media Blender.

Complete descriptions of all the properties available in blender.properties appear in the Media Blender Reference Guide. At installation, you should be concerned only with identifying the participating media in your configuration.

Use the Medium(*n*) property in blender.properties to identify all participating media in your configuration.



Identify the properties files for all media used with this Media Blender. When specifying media, number each medium sequentially, beginning with 1. In addition, be sure to list the queuing medium last. (In most cases, this will be Collaboration.properties.) Note that these properties files must exist in the same directory as blender.properties.

Example:

```
medium1=ACD.aspect.properties  
medium2=Collaboration.properties
```

Note: You can configure Media Blender to automatically start whenever the JWS is started (using the autostart property in your Blender properties file.) Be sure not to set this autostart property to true until you are sure that all of your media are properly configured. See Post-Installation Tasks, later in this document.

The next step is to configure the properties files you just identified in the medium property.

Configure the connection to Media Blender

Configuration properties files for participating media contain properties that ensure proper communication between the Media Blender and both the ACD and the Collaboration medium.

In this section:

- [Collaboration medium](#)
- [Aspect medium](#)
- [ASAI medium](#)
- [CT-Connect medium](#)
- [JTAPI medium](#)
- [Lucent ICM medium](#)
- [Media Blender Telephony Integration API](#)

Collaboration medium

The following properties are included in Collaboration.properties, which is the properties file for the Collaboration medium. This file typically resides in <weblines dir>/servlet/properties/Blender.

Use these properties to configure the Collaboration medium's connection to Media Blender:

wlapi_host=

Property type: string

Default value: none

Enter the host name or IP address of the WCS.

Important: This property must be entered in all lowercase letters.

wlapi_port=

Property type: integer

Default value: none

Specify the port Media Blender uses to connect to WCS. Typically, this is port 80.

Important: This property must be entered in all lowercase letters.

Aspect medium

The following properties are included in ACD.aspect.properties, which is the properties file for the Aspect medium. This file typically resides in <weblin dir>/servlet/properties/Blender.

Use these properties to configure the Aspect medium's connection to Media Blender:

socketport=

Property type: Integer

Default value: None

This property identifies the port number used in the Application Bridge setup.

linkid=

Property type: Integer

Default value: None

This property identifies the link ID used in the Application Bridge setup.

delimiter=

Property type: Character

Default value: , (comma)

This property identifies the delimiter used in the Application Bridge setup.

header=

Property type: String

Default value: None

This property identifies the host name used in the Application Bridge setup. Be sure to use correct capitalization for the host name. The Aspect system requires an exact match.

appid=

Property type: Integer

Default value: 0

This property identifies the Aspect application ID.

ASAI medium

The following properties are included in ACD.asai.properties, which is the properties file for the ASAI medium. This file typically resides in <weblines dir>/servlet/properties/Blender.

Use these properties to configure the ASAI medium's connection to Media Blender:

hostname=

Required. This property contains the IP address or network name of the Multi-Application Platform (MAPD) on the Definity switch. See the *Media Blender Switch Administrator's Guide: Lucent Definity Generic 3* for more information.

link=

Required. This property indicates the Link number used on the MAPD board. This number must match the value in the **Link** field in the DLG Administration field on the Lucent Definity. See the *Media Blender Switch Administrator's Guide: Lucent Definity Generic 3* for more information.

expertagentselection=

Property type: Boolean

Default value: true

This property indicates whether the switch supports Expert Agent Selection (EAS).

CT-Connect medium

The following properties are included in ACD.ctc.properties, which is the properties file for the CT-Connect medium. This file typically resides in <weblines dir>/servlet/properties/Blender.

Use these properties to configure the CT-Connect medium's connection to Media Blender:

servername=

Property type: String

Default value: none

Required. This property identifies the network name or IP address for the CT-Connect server.

logicalidentifier=

Property type: String

Default value: None

Required. This property identifies the name for the link to the ACD at the CTC Server.

networktype=

Property type: String

Default value: ncacn_ip_tcp

This property identifies the type of network used.

Network Protocol	Value
NetBios over NetBEUI	ncacn_nb_nb
TCP/IP	ncacn_ip_tcp
DECnet	ncacn_dnet_nsp
NetBIOS over TCP/IP	ncacn_nb_tcp
Named pipes	ncacn_np
Novell SPX	ncacn_spx

protocol=

Property type: String

Default value: CSTA

This property identifies the switch protocol.

Acceptable values are:

Value	Switch Protocol	ACDs
CSTA	Computer Supported Telephony Applications	Switches with CSTA-compliant links that are supported by Dialogic CT-Connect
ASAI	CallVisor ASAI G3	Lucent Definity Generic 3
MLP	Meridian Link Protocol	Nortel Meridian 1
MLS	Meridian Link Service	Nortel Symposium

agentpasswordrequired=

Property type: Boolean

Default value: false

This property indicates whether the switch requires passwords for agent login.

Acceptable values: true, false

CSTA and Nortel Switches only:

preassignagents= (used for CSTA and Nortel switches only)

Property type: Boolean

Default value: false

This property indicates whether the CT-Connect medium assigns CTC channels to agents during Media Blender initialization. Setting this property to true enables unblended login for CSTA and Nortel switches.

agentidrequired= (used for CSTA and Nortel switches only)

Property type: Boolean

Default value: true

This property indicates whether the switch requires an agent ID for logging in. (This property is not applicable to the Lucent Definity G3; for this switch, make sure to set the expertagentsselection property appropriately.)

CSTA switches:

ctstaphase= (used for CSTA switches only)

Property type: Integer

Default value: 1

This property defines the CSTA protocol level supported by the switch. This property is required only if the protocol property is set to CSTA.

Acceptable values: 1, 2

Nortel switches:

positionidfile=(used for Nortel switches only)

Property type: String

Default value: positionids.properties

This property identifies the name of the file that maps Nortel telephone extensions to position IDs.

To identify Expert Agent Selection (EAS) (Lucent Definity switches only):

expertagentselection=

Property type: Boolean

Default value: false

This property indicates whether the switch supports Expert Agent Selection (EAS). This property is applicable only if the protocol property has been set to ASAI for the Lucent Definity G3 switch.

Acceptable values: true, false

JTAPI medium

The following properties are included in ACD.jtapi.properties, which is the properties file for the JTAPI medium. This file typically resides in <weblinedir>/servlet/properties/Blender.

Use these properties to configure the JTAPI medium's connection to Media Blender:

peer=

Property type:String

Default value:none

This property identifies the JTAPI implementation used by Media Blender. For Lucent Passageway, this should be set to com.lucent.jtapi.tsapi.TsapiPeer.

user =

Property type: String

Default value: none

This property is the username to log in to JTAPI. For Lucent Passageway, this value should be a username on the PassageWay TServer. The user must have privileges to access call control and monitoring services on agent terminals that receive calls from Media Blender.

password =

Property type: String

Default value: none

This property is the password for the JTAPI user listed above.

service =

Property type: String

Default value: none

This property indicates the CSTA Tlink to the ACD/PBX on the Passageway TServer. For Lucent Passageway, this is typically LUCENT#<switchname>#CSTA#<tserver name>. (See your switch administrator for more information.)

Lucent ICM medium

The following properties are included in ACD.icm.properties, which is the properties file for the Lucent ICM medium. This file typically resides in <weblines dir>/servlet/properties/Blender.

Use these properties to configure the Lucent ICM medium's connection to Media Blender:

host=

Property type: String

Default Value: None

This property specifies the host name or IP address of Lucent Internet Call Manager (ICM) server.

Note: If you are using the same machine to run both the ICM server and Media Blender, you must set up the Java Web Server to run on a port other than port 80. See [Changing the Java Web Server Port](#) for more information.

Media Blender Telephony Integration API

The following properties are included in `ACD.socket.properties`, which is the properties file for the Media Blender Telephony Integration API (MBTI API). This file typically resides in `<weblinedir>/servlet/properties/Blender`.

Use these properties to configure the MBTI API medium's connection to Media Blender:

socketport=

Property type: integer

Default value: 7002

This property identifies the port number used by MBTI API. The default port number is 7002.

mediumisserver=

Property type: Boolean

This property determines if this ACD medium acts as a MBTI API or a MBTI API client.

sockethost=

Property type: String

This property is used only if the `mediumisserver` property is set to false. This property indicates the name of the server computer where the socket server application is running.

premonitoragents=

Property type: Boolean

Default value: true

This property indicates whether agents are monitored during Media Blender initialization. If it is set to true, all agents listed in the file specified by the `physicallocationfile` property are monitored. This property should be set to true if you are using unblended agent signon.

monitoracds=

Property type: Boolean

Default value: false

This property indicates whether ACDs are monitored for queue events from `BlenderSocket`.

ignoreextraparameters=

Property type: Boolean

Default value: false

This property lets you determine whether to ignore unrecognized reply and event message parameters. If set to true, unrecognized reply and event message parameters are ignored and the message is accepted. If set to false, unrecognized parameters cause reply or event messages to be rejected and an error message appears in the Blender log file.

Establish call types and outbound dialing strategies

Blended Collaboration configurations can provide different types of callBack to customers. For instance, you can ensure that agents can converse with callers via voice calls, chat sessions, or voice-over-IP calls. Media Blender obtains a *call type* from the callBack form (callBackACD.html) submitted by the caller and uses different outbound callflows (*CTI strategies*) when providing callBack to customers.

In this section:

- [Strategies for different call types](#)
- [Mapping call types to CTI strategies](#)
- [Identifying the calltype file to your ACD medium](#)
- [Define a Pool of Phantom Lines](#)
- [If you are using only one calltype](#)

Note: This section is not applicable to the Lucent ICM driver.

Strategies for different call types

The available CTI strategies have been designed to provide appropriate callBack in different configurations and for different call types. This section describes those that should be used for each call type.

Note: See [About Media Blender](#) for an overview of outbound dialing strategies.

Strategies for Voice calls

Use any of the following three strategies for voice calls:

Predictive

The ACD places an outbound call to the Web caller. The ACD connects to the agent only after reaching the caller.

Use the Predictive strategy only if you can assure with some certainty that an agent will always be available. With this strategy, if an agent is not available, the caller will be put on hold immediately after receiving the call.

PhantomWaitRelease

Media Blender dials into an queue using one of the phantom lines. Once the agent answers, the phantom line is placed on hold while Media Blender places an outbound call to the caller using the agent's second line. Once the caller answers the phone, the phantom line is released. Note that while the phantom line is on hold, no other call can be routed to the agent.

PhantomWaitNoCallRelease

This strategy is similar to PhantomWaitRelease except that the phantom stays connected to the agent for the length of the call. This provides more accurate reporting from the ACD, but requires more phantom lines.

Strategies for chat sessions

Use either of these strategies for chat calls:

PhantomNoCallRelease

Use this strategy if you want to provide chat sessions and if your ACD can place agents in a *busy* state as soon as an their phones disconnect from another. PhantomNoCallRelease connects to an agent but releases the phantom line immediately. The agent is placed in a busy state, allowing the agent and caller to engage in a text chat session uninterrupted.

The PhantomNoCallNoRelease

Use this strategy if you want to provide chat session and if your ACD does not support the automatic *busy*ing out of agents. You should always use this strategy if you require more accurate reporting, you should use this strategy. Media Blender uses the phantom line to connect to the agent; however, the phantom doesn't release the agent's phone until the session is complete. This provides more accurate reporting, but requires a larger phantom pool.

Mapping call types to CTI strategies

You establish a text file that maps the web-based call type code from the callBack page to the CTI strategies described in the previous section. This file is typically called calltypes.properties file and resides in the <weblinedir>/servlet/properties/Blender directory.

Each line of the call types table maps web-based textual call type codes embedded in the callBack page to CTI Strategies in the following format:

```
call type code=<CTI Strategy>
```

where

- *call type code* is a code embedded in a field on the WebLine CallBack HTML form used by the caller. (We provide a sample callback form, \<weblinedir>\pub\html\forms\callFormACD.html, that your WCS administrator can use to create the callback form for your site.
- *CTI Strategy* defines the strategy to be followed by the call type. Acceptable values are:

Predictive

PhantomWaitRelease

PhantomWaitNoRelease

PhantomNoCallRelease

PhantomNoCallNoRelease

Von

See [Strategies for different call types](#) for complete descriptions of these values.

Defining a Default Call Type

Be sure that the calltypes.properties file defines a default call type. The entry should appear as follows:

```
default=<ctistrategy>
```

Note: Call type matching is not case sensitive.

Example calltypes.properties file

This example maps call type codes to CTI Strategies:

```
pstn=Predictive
```

```
chat=PhantomNoCallNoRelease
```

```
default=Predictive
```

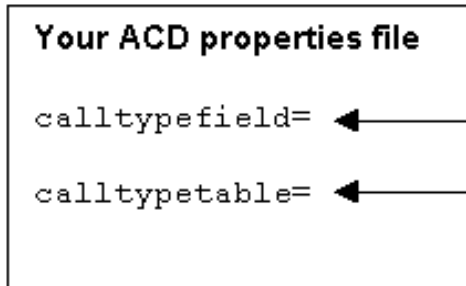
operator=PhantomWaitRelease

von=Von

Identifying the calltype file to your ACD medium

Your ACD properties file identifies:

- the field on the callback page used to determine call type
- the name of the calltype file



Enter the name of the field on the callback page that contains the call type.

Enter the name of the file maps call type codes to CTI strategies.

About the calltype field

The call type can be passed in any of the *Appl* fields on callFormACD.html. In the `calltypefield` property, determine which of these fields should contain the call type code as follows:

If you are using WCS release 3.0:

Enter this value in the <code>calltypefield</code> property...	...to indicate this <code>Appl</code> field on callFormACD.html
USER1 (default)	Appl1

Note that, with WCS 3.0, you can also define a unique calltype variable on the callback page. If you do so, be sure to point to the correct variable using the `calltypefield` property in your ACD properties file.

For WCS release 2.51:

Enter this value in the <code>calltypefield</code> property...	...to indicate this <code>ApplStr</code> field on callFormACD.html
USER1 (default)	Appl1
USER2	Appl2
USER3	Appl3
USER4	Appl4

Make sure you must enter the `calltypefield` property values using all uppercase letters.

For more information on setting up your callback page, see [Tailor your callBack page](#) in [Section 3: Advanced/Optional Configuration](#).

Define a Pool of Phantom Lines

To use any of the phantom line CTI strategies, your switch administrator needs to set up a pool of phantom lines. Phantom lines wait in the queue on behalf of the caller when predictive calling is not possible. (See [Strategies for different call types](#) for more information about Phantom line CTI strategies.)

Note: For the Lucent Definity G3 switch, phantoms can be either physical or virtual phones. On all other systems, the phantom line **must** be a physical telephone. See the *Media Blender Switch Administrator's Guide: Lucent Definity Generic 3* for more information.

When setting up this phantom pool, you need to provide some information to the Switch Administrator, who in turn must provide information to you. The tasks involved are:

- [Determining how many phantom lines are needed at your site](#). Once you've determined how many lines you need, provide this information to the Switch Administrator.
- [Setting up a file to identify the phantom lines](#). After the pool has been established, you need to obtain phantom line IDs from the Switch Administrator.
- Identify the phantom pool file to your ACD medium

Determine Phantom Line Requirements

Use this formula when determining how many phantom lines are needed on your system:

number of phantom lines needed = $c * s / 3600$

where:

- c represents the number of blended calls per hour
- s represents the seconds per phantom usage. When using the PhantomWaitRelease strategy, this is the average queue time. When using the PhantomWaitNoRelease strategy, this is the average queue time plus average talk time.

Example:

This example assumes that the CTI strategy used is PhantomWaitRelease. There are 3600 calls per hour and the queue time per call is equal to 10 seconds.

- number of phantom lines needed = $3600 * 10 / 3600$
- number of phantom lines needed = 10 lines

To identify each phantom line used at your site, you must:

- Set up a file listing each phantom line and its type. This file is typically named `phantoms.properties` and must be located in the `<weblinedir>\servlet\properties\blender` directory.
- Identify the phantom pool properties file to your ACD medium using the `phantompool` property in the properties file for your ACD medium.

Setting up a phantom pool file

The `phantoms.properties` lists the phantom lines and their types in the following format:

phantomID=phantom type

where

- *phantomID* is either the phantom's physical ID on the ACD or an agent's permanent extension.
- *phantom type* identifies the line type. Note that Media Blender currently supports only digital line types. Type *D* to indicate digital lines.

Example:

12345=D

67890=D

23456=D

Using agent logical IDs in phantoms.properties

Some ACDs, such as Aspect, require a logged in agent ID before it will place calls. In this case, you can use agent permanent extensions (or logical IDs) in place of the phantomIDs in the phantoms.properties file.

Follow these steps to use permanent extensions when creating your phantom line pool:

1. Create phantoms.properties using agent permanent extensions in place of phantom ID. For example:

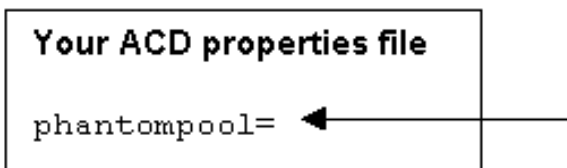
```
agent_permanent_extension=phantom type
```

2. Create an entry for the phantom's logical ID in the agents.properties file.
3. Create an entry for the phantom's password in the password.properties file.

See [Set up agent log in](#) for information about setting up agents.properties and password.properties.

Identify the phantom pool file to your ACD

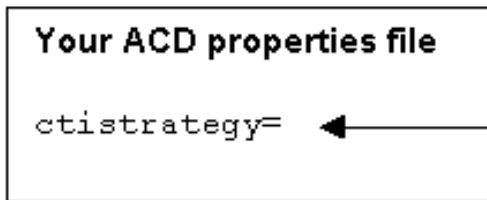
Use this property (in the properties file for your ACD medium) to identify the phantom pool file:



Enter `phantoms.properties`, or whatever name you assigned to the file that lists phantom extensions.

If you are using only one calltype

If you don't need to accommodate different types of callBack, you need not create a call type table to specify the appropriate CTI strategy. Instead, simply specify the CTI strategy used for the type of call you are providing in your ACD properties file.



Identify the outbound call flow you want to use with the type of callBack you are providing. Possible values are:

Voice calls:

Predictive
PhantomWaitRelease
PhantomWaitNoRelease

VoIP calls:

Von

Text chat:

PhantomNoCallRelease
PhantomNoCallNoRelease

Set up agent log in

In Blended Collaboration configurations, agents must log in to their ACD phone system as well as to the Collaboration server.

To log in to a phone system, agents typically need to enter:

- Physical IDs
- Agent Passwords

Physical IDs are unique for each phone. All agents who use a particular phone must use the same physical ID and password.

To log in to the WCS, agents enter their username and *permanent extension* (sometimes referred to as the agent's *logical ID*).

Important: The agent's permanent extension MUST match the logical ID of the agent on the ACD.

Agent log in options

You have these options when setting up agent log in:

- [Unblended log](#) in Agents log in to WCS and to the ACD separately.
- Blended log in--Agents log in to the WCS and ACD simultaneously. The blended login method you use depends on whether your agents:
 - always use the same phone
 - frequently use different phones

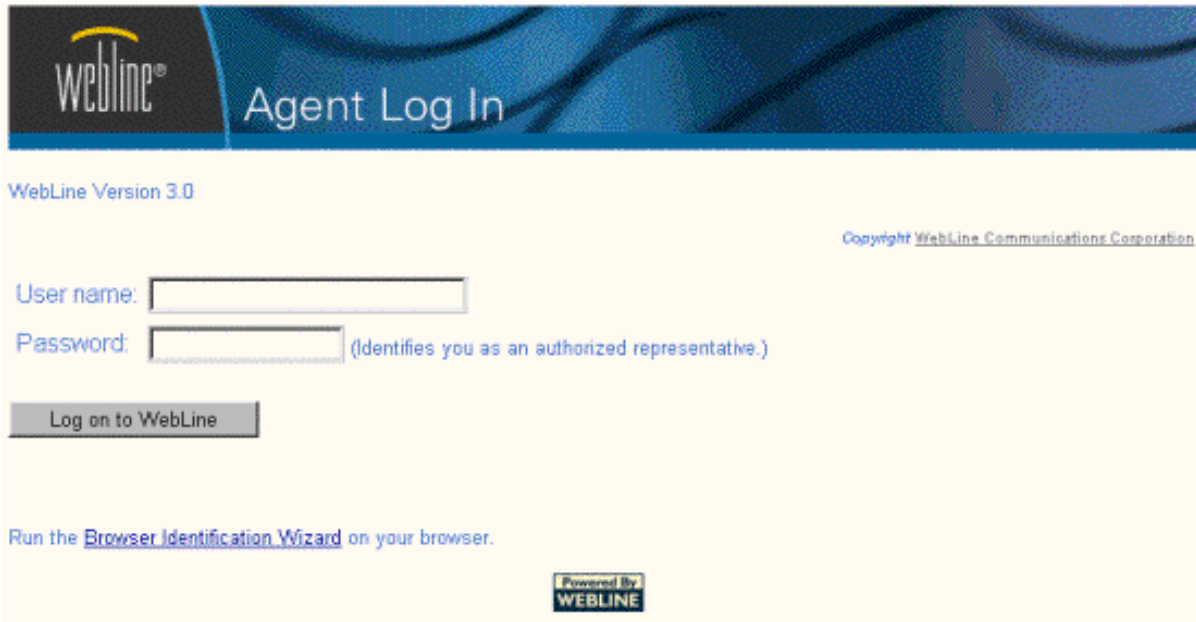
Unblended log in

To achieve unblended agent login, the agent logs in to WebLine using the standard WebLine agent log on page (<weblinedir>\pub\html\forms\agentLogon.html). To ensure unblended log on, you must set the acceptsignon property to false in the properties file for your ACD medium.

Note: You cannot use unblended agent signon with Lucent ICM media and most Dialogic CT-Connect media. You can achieve unblended log on Lucent Definity CT-Connect switches using the monitorskills property. See the CT-Connect section of the *Media Blender Reference Guide* for more information on the monitorskills property). Unblended login works correctly with Aspect and ASAI media.

Blended log in: If agents always use the same phone

You can set up *blended login* if your agents always use the same phone. Agents sign on to WebLine Collaboration using the standard agent log in page (<weblinedir>\pub\html\forms\agentLogon.html, which resides on the WCS). When they log in, they are automatically logged into the phone system as well.



To achieve this scenario, you must link the agent's WCS permanent extension (or logical ID) with physical phone IDs and passwords :

- [Set up agents.properties](#)--This file maps each agent's logical ID to a physical phone ID. This file must reside in <weblinedir>\servlet\properties\blender directory.
- [Set up passwords.properties](#)--This file maps each agent's logical ID to a password. This file must reside in the <weblinedir>\servlet\properties\blender directory. For phone systems that do not require agent passwords, this step can be skipped.
- [Identify these properties file to your ACD medium](#) using the physicallocationfile property and the passwordfile property of your ACD's properties file.

Set up agents.properties

You can map an agent's logical ID to a specific physical phone ID using the Physical Location File. (This file is typically named agents.properties and must be located in the <weblinedir>\servlet\properties\blender directory.) For each agent you want to add, include a line in this file that lists the agent's logical ID and its corresponding phone physical ID in this format:

```
agent's logical ID=physical ID
```

Example:

```
45017=40017
```

```
45023=40032
```

Note: In order to enable changes to this properties file, Media Blender must be restarted. See online help for the Media Blender Control panel for more information.

Set up passwords.properties

You can assign a password to an agent's logical ID using the Password file. (This file is typically named password.properties and must be located in the <weblinedir>\servlet\properties\blender directory.)

The Password File maps an agent's logical ID to a specific password. For each agent you want to add, include a line in this file that lists the agent's logical extension and his or her corresponding password in this format:

```
agent's logical ID= password
```

Example:

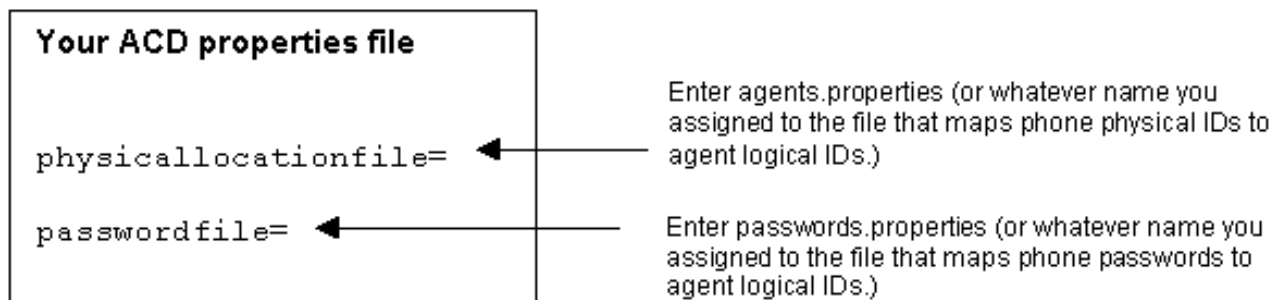
```
45017=32363
```

```
45023=32332
```

Note: In order to enable changes to this properties file, Media Blender must be restarted. See online help for the Media Blender Control panel for more information.

Identify physical location and password files to your ACD medium

Use these properties (in your ACD properties file) to identify the physical location and password files:



Note: If you are using a Nortel switch and want to achieve blended log in, you must create an agent group file. See [Create an Agent Group File \(CSTA switches only\)](#) in [Driver-specific setup tasks](#) for more information.

Blended log in: If agents frequently use different phones

You can set up blended dynamic log in if your agents do not consistently use the same phones. Agents use a WebLine agent log in page designed specifically for log on to an ACD. The HTML form `<weblinedir>\pub\html\forms\agentLogonACD.html` allows agents to specify their phone physical IDs and passwords when logging in.

weblined[®] Agent Log In

WebLine Version 3.0 Copyright WebLine Communications Corporation

User name:

Password: (Identifies you as an authorized representative.)

Terminal ID:

Switch Password:

Run the [Browser Identification Wizard](#) on your browser.

Powered by
WEBLINE

Using this log in page ensures that the agent must enter the terminal ID and password when logging on to WebLine Collaboration Server. That way, agents can log on from different stations and always use the phone next to the computer.

To achieve blended dynamic log on, you need to ensure that Media Blender does not look to the `agents.properties` and `password.properties` file to retrieve terminal IDs and password. You can do so using either of these methods:

- Comment out or remove the `physicallocationfile` and `passwordfile` properties from your ACD medium's properties file
- Delete all entries in `agents.properties` and `password.properties`

Note: If you are using a Nortel switch and want to achieve blended log in, you must create an agent group file. See [Create an Agent Group File \(CSTA switches only\)](#) in [Driver-specific setup tasks](#) for more information.

Set Up Skills-Based Routing

Media Blender ensures web-based callback requests are routed to appropriate agents by mapping to the routing capability of your ACD.

Before you begin, ensure that your ACD is configured to route web calls to agents who have access to the Collaboration Server. In addition, ensure the switch will route calls to agents who will handle each call type. For instance, if all of your web-enabled agents will receive voice calls, but only some will receive chat, ensure the switch is configured to route chat calls to those agents.

See the *Media Blender Switch Administrator's Guide* for your ACD for more information about setting up routing addresses.

You must obtain the appropriate routing addresses from your switch administrator to complete the steps in this section.

In this section:

- [Map Web-based skills to ACD logic](#)
- [Identify your skill table to your ACD medium](#)

Map Web-based skills to ACD logic

Media Blender obtains a web-based routing code from the callback form used by the caller to request call back . You can map these routing codes to skill groups set up on the switch by creating a skills table, typically called skills.properties. (This file typically resides in <weblinedir>/servlet/properties/Blender.)

Each line of the skills table maps the WebLine Routing address to the routing logic used by your ACD, in the following format:

Routing Address=routing number

where

- Routing Address is a code embedded in the WebLine Callback HTML form used by the caller. (The Routing Address is set in a hidden field, RoutingAddr, on the Callback form. We provide a sample callback form, /<weblinedir>/pub/html/forms/callFormACD.html, that your WCS administrator can use to create the callback form for your site.)
- Routing number is equal to a value unique to the routing logic on your ACD. Each ACD type uses different routing logic, as follows:

ACD type	Routing Number
Aspect	CCT
CSTA	RPN
Lucent	VDN
Meridian	CDN

See your switch administrator to obtain appropriate routing numbers. In most cases, it will be necessary to create a new routing number (i.e. CCT, VDN, and so forth) on the switch for use with the WebLine application. Refer to the WebLine Media Blender switch-specific documentation and consult your switch administrator for more information.

When a call back request comes in, Media Blender retrieves the routing address from the callback form and matches it to the ACD routing number. The ACD then routes the request to the appropriate agent .

Define a Default Route

Be sure that the skills.properties file defines a default route, equivalent to the default routing number used on the switch. The entry should appear as follows:

default=<routing number>

Example skills.properties file

This example maps a WebLine Routing address to an Lucent VDNs:

```
default=9520
```


sales=9520

westernsales=9530

easternsales=9540

collections=9550

Important: When agents are NOT logged in to WCS

Keep in mind that, although Media Blender can ensure that web-initiated calls are routed to web-enabled agents, Media Blender CANNOT verify whether an agent who receives a routed call is actually logged in to the Collaboration server.

You must ensure that the routing numbers listed in skills.properties route calls to agents who are logged in to the Collaboration server. The following describes what happens when an agent is logged in to the ACD, but NOT to the WCS:

- The ACD will route web-initiated calls to that agent, along with all other agents in that agent group.. The agent and caller are able to converse on the telephone.
- The WCS will NOT be able to assign that call to that agent
- The WCS session will be cancelled and the caller will receive a *Session Terminated* page.

Since the agent and caller can talk over the phone, the agent can provide a Collaboration session using meetMe routing, rather than blended callBack.

Identify your skill table to your ACD medium

Use the following property (in your ACD properties file) to identify the skills table:

Your ACD properties file
`skilltable=` ←

Enter the name of your skills table file.

Driver-specific setup tasks

This section contains information you need when configuring connectivity to your particular switch/CTI driver.

In this section:

- [Aspect Media: Configure Eventbridge](#)
- [CT-Connect driver](#)
- [Lucent ICM driver](#)

Aspect Media: Configure Eventbridge

If your Aspect ACD has EventBridge software, you can choose to use it with Media Blender or not. If you want to use EventBridge, you must set up a properties file that specifies how each agent or agent group should be monitored. To do so, you must:

- Set up an EventBridge properties file . This file is typically named eventbridge.properties and must reside in the <webLine dir>\servlet\properties\blender directory.
- Identify this properties file to your Aspect ACD medium using the eventbridgemap property in the ACD.aspect.properties file.

The eventbridge.properties file lets you monitor the status of agent groups and their calls. Each entry in the file defines an EventBridge map. Each map identifies a particular agent group and defines how their calls should be monitored. Each EventBridge map entry should appear in the following format (with spaces separating each parameter):

```
Map#= whom number agentStateMap callStateMap
```

where:

- *Map#* identifies the map. The eventbridge.properties file can have any number of maps, defining how many agent groups should be monitored.
- *whom* identifies the agent group.

Acceptable values are:

Value	Description
AG	Monitors a specific agent group
ASG	Monitors a specific agent supergroup
AAG	Monitors all agent groups (i.e. all agents)
TG	Monitors a specific trunk group
ATG	Monitors all trunk groups

- *number* is the number of the agent group defined in whom.
- *agentStateMap* defines the agent states that should be monitored. The value you enter here consists of 10 Boolean characters (Y or N). Each position in agentStateMap let you monitor a different agent state, as follows:

Position	Agent State Event
0	Agent logged on event
1	Agent logged off event
2	Agent idle event
3	Agent available event
4	Agent wrap-up event
6-9	Reserved for later use (set as N)

- *callStateMap* defines the call states that should be monitored. The value you enter here consists of 20 Boolean characters (Y or N). Each position in agentStateMap let you monitor a different call state, as follows:

Position	Call State Event
0	Call noticed event
1	Call offered event
2	Call connected event
3	Call queued event
4	Call dequeued event
5	Call held event
6	Call retrieved event
7	Call transferred event
8	Call conferenced event
9	Call disconnected event
10-19	Reserved for later use, set as N

Example:

Map1= AG 1 YYYYYYNNNN YYYYYYYYYYNNNNNNNNNN

CT-Connect driver

To ensure proper connectivity between CT-Connect and Media Blender, complete the following tasks:

- [Determine CT-Connect Monitor Requirements](#)
- [Install the CT-Connect API](#)
- [Create an Agent Group File \(CSTA switches only\)](#)
- [Create a Position ID file \(Nortel switches only\)](#)

Determine CT-Connect Monitor Requirements

Some of the telephony devices used in Media Blender configurations require more than one CT-Connect monitor. For instance, agents and phantom phones use two CT-Connect monitors. Therefore, an unlimited Dialogic CT-Connect monitor license is strongly recommended. You determine the maximum number of CT-Connect monitors when you configure the link from your CT-Connect server to your switch.

The following sections describe how to determine how many CT-Connect monitors are necessary in different configurations:

Lucent Definity--Predictive CTI Strategy

1
+ Number of unique VDN extension numbers listed in skills.properties
+ 2 x the number of logged-in and/or pre-assigned* agents
+ 2x the number of skill group extensions specified in the monitorskills property in ACD.ctc.properties
= Required number of CT-Connect monitors

Lucent Definity using a phantom CTI Strategy

1
+ 2 x the number of phantom lines
+ 2 x the number of logged-in and/or pre-assigned* agents)
+ 2 x (# of skill group extensions specified in the monitorskills property in ACD.ctc.properties

Nortel Meridian or Nortel Symposium

1
+ 2 x the number of phantoms
+ 4 x the number of logged-in and/or pre-assigned* agents
= Required number of CT-Connect monitors

CSTA Switches--Predictive CTI Strategy

1
+ 2 x the number of unique ACD extension numbers listed in skills.properties
+ 2 x the number of logged-in and/or pre-assigned* agents
= Required number of CT-Connect monitors

CSTA Switches--Phantom CTI Strategy

1

+ 2 x the number of phantom lines

+ 2 x the number of logged-in and/or pre-assigned* agents

= Required number of CT-Connect monitors

.
***Note:** Agents are preassigned during Media Blender startup if the preassignagents property in ACD.ctc.properties is set to true. For the Lucent Definity G3 and for all CSTA switches, all agents listed in agents.properties are assigned channels and monitored. For Nortel switches configured with agent IDs (agentidrequired=true), all agents listed in agents.properties are assigned channels and monitored.

For Nortel switches not configured with agent IDs (agentidrequired=false), all agents listed in positionids.properties are assigned channels and monitored.

If preassignagents is set to false, the channel assignment and monitoring for any given agent does not occur during startup, but only when that agent first logs in.

Install the CT-Connect API

Install the CT-Connect API

After installing Media Blender, follow these steps to ensure correct installation of the CT-Connect API:

1. Install the CT-Connect API on the same computer on which Media Blender will run. The CT-Connect API resides on the Dialogic CT-Connect Server installation CD. Refer to the *CT-Connect for Windows NT Installation and Administration Guide* for details about installing the CTC API. (After installing the CTC API, you should see the file `ctcapi32.dll` listed in the `C:\WINNT\system32` folder.)
2. Manually copy the file `JniCtc.dll` from `C:\<weblines dir>\bin` to `C:\WINNT\system32`.

Create an Agent Group File (CSTA switches only)

Some switches require agents to enter their agent group numbers when logging in. To achieve blended login (as described in [Set up agent log in](#)) on these switches, you must create an agent group file to map agent IDs with agent groups. This file is typically called groups.properties and must reside in the `<webline dir>\servlet\properties\blender` directory.

After you create the groups.properties file, identify it in your ACD medium's properties file using the groupfile property.

Create groups.properties

Each line of the groups file maps agent IDs to groups set up on the switch, in the following format:

```
agent ID=agent group
```

Define a Default Group

Be sure that the groups.properties file defines a default agent group, equivalent to the default VDN. The entry should appear as follows:

```
default=<agent group>
```

Example groups.properties file

The groups.properties file maps agent logical IDs to agent groups set up on the switch. Below is a sample group file.

```
default=4320  
100=4320  
101=4320  
102=4320  
103=4320  
104=4320  
105=4320  
106=4321  
107=4321  
108=4321  
109=4322  
110=4322  
111=4321 4322
```

Create a Position ID file (Nortel switches only)

When setting up a Nortel switch, you must create a position ID file, which maps Nortel telephone extensions to position IDs. This file is typically called `positionids.properties` and must reside in the `\<webline dir>\servlet\properties\blender` directory.

After you create the `positionids.properties` file, identify it in your ACD medium's properties file using the `positionidfile` property.

Note: This file is applicable to Nortel switches only.

Create `positionids.properties`

Each line of the `positionid` file maps the Nortel telephone extensions to position IDs in the following format:

Nortel telephone extension=Nortel position ID

Example `groups.properties` file

340=240

341=241

342=242

343=243

Lucent ICM driver

To ensure proper connectivity between Lucent ICM and Media Blender, you must understand some considerations regarding where Lucent ICM is installed.

You have the option of installing the Lucent ICM software and Media Blender on the same machine or on separate machines. The next sections summarize steps you must take for either configuration.

Note: Installing Media Blender and Lucent ICM on the same machines is not recommended.

If Lucent ICM and Media Blender are on the same machine

If you've installed Media Blender on the same machine as the Lucent ICM software, you need to change the port used by the Java Web Server.

To change the JWS Port

1. Edit this file :

```
<JWS Install  
Dir>\properties\server\javawebserver\webpageservice\endpoint.properties
```

2. Set the endpoint.main.port property to a port other than 80 (e.g. 8080).

If Lucent ICM and Media Blender are on different machines

If you've installed Media Blender and Lucent ICM software on separate machines, you need to copy the ITG class files to the Media Blender machine and ensure that the CLASSPATH includes these files. Follow these steps:

1. Create the directory c:\itg on the Media Blender machine

2. Copy the com directory from the ICM server into c:\itg on the Blender machine. The destination directory should now be c:\itg\com.

3. Add c:\itg to your CLASSPATH on the Media Blender machine. (The CLASSPATH should be

```
c:\weblinedir>\servlet;<weblinedir>\servlet\wlapi.jar;c:\itg). See Setting the CLASSPATH Variable for more information.
```


Section 3: Advanced/Optional Configuration

This section describes optional features you can set up in your Blended Collaboration environment. It includes these sections:

- [Setting up your log file](#)
- [Configure the Collaboration Server](#)
- [Set up email notification](#)
- [Using Multiple Media Blenders](#)

Setting up your log file

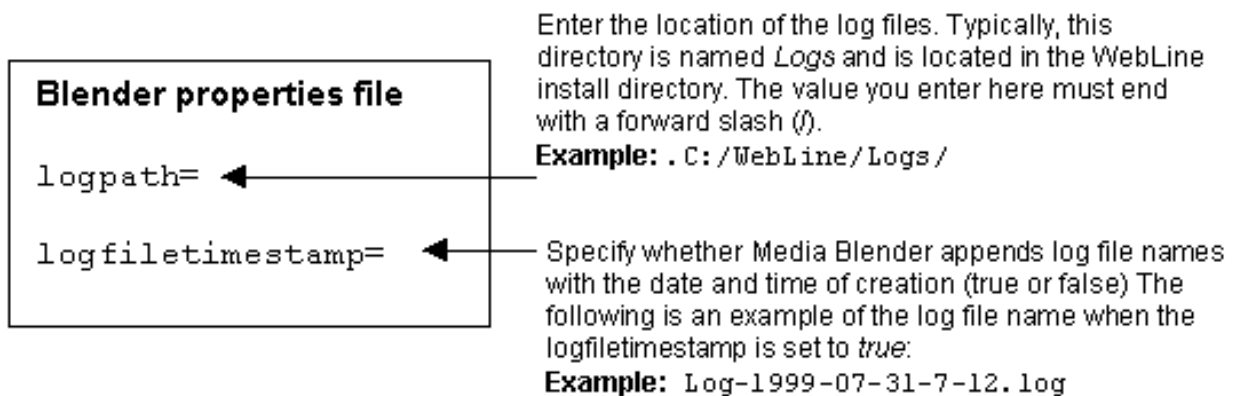
Media Blender logs alerts and messages into a log file. Several properties in the Blender properties file determine how the log file is named and where it resides.

You can specify:

- [Path and file name](#)
- [A maximum number of log files](#)
- [The logging level](#)

Specify path and filename

The following properties let you specify where the logfile resides, as well as a naming convention for the filename.



Blender properties file

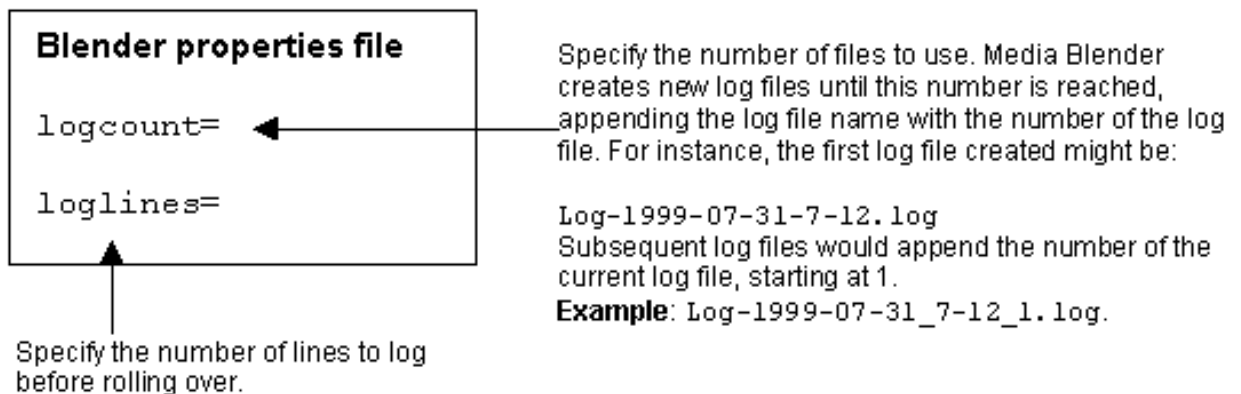
```
logpath=  
logfiletimestamp=
```

Enter the location of the log files. Typically, this directory is named *Logs* and is located in the WebLine install directory. The value you enter here must end with a forward slash (/).
Example: .C:/WebLine/Logs/

Specify whether Media Blender appends log file names with the date and time of creation (true or false) The following is an example of the log file name when the logfiletimestamp is set to *true*:
Example: Log-1999-07-31-7-12.log

Specifying maximum number of logs

The following properties let you specify a maximum number of log files. Note that if you want only one log file to be created, you should set the logcount property to 1.



Blender properties file

```
logcount=  
loglines=
```

Specify the number of files to use. Media Blender creates new log files until this number is reached, appending the log file name with the number of the log file. For instance, the first log file created might be:
Log-1999-07-31-7-12.log
Subsequent log files would append the number of the current log file, starting at 1.
Example: Log-1999-07-31_7-12_1.log.

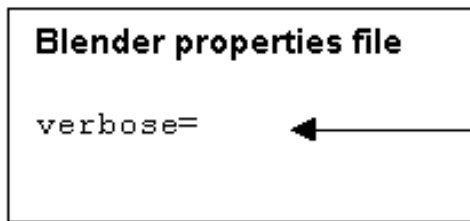
Specify the number of lines to log before rolling over.

Setting the logging level

This property in blender.properties lets you specify the kinds of alerts you want to include in the log files:

Specify logging level for the log file. The levels indicate what is included in the logfile, as follows:

```
Blender properties file  
  
verbose=
```



1—Urgent. The log includes only urgent messages and errors

2 – Critical. The log includes critical and urgent messages, and errors

3 – Important. The log includes urgent, critical and important messages, and errors

4 – Informational. The log includes informational, urgent, critical, and important messages, as well as errors.

Configure the Collaboration Server

On the WCS, you must:

- [Set up the Blender Agent](#)
- [Tailor your callBack page](#)
- [Turn off meetMe AutoConnect](#)

Set up the Blender Agent

On the WebLine Collaboration Server (WCS), you set up a *Blender Agent*, which is the agent used by Media Blender to log in to the WCS. The Blender Agent is set up in the agent database like any other agent, with a username, password, and permanent extension. On the Blender Server, you also identify the Blender Agent using the loginid and loginpw properties in the WCS properties file (Collaboration.properties).

Note: Do not confuse the *Blender Agent* with actual WCS agents who sign onto the switch and the WCS and take calls.

The role of the Blender Agent's permanent extension

You use the Blender Agent's permanent extension to direct callBack requests to the Media Blender. This permanent extension is embedded in the callback request form served to the caller when requesting call back. When the caller fills out and returns the form, the request is routed via the permanent extension number to Media Blender.

The Blender Agent's permanent extension is set to 125 by default on the WCS. (You can change this number, if you wish.) This permanent extension also appears in our sample callback HTML page, callFormACD.html (located in <weblines dir>/pub/html/forms directory on the WCS).

Tailor your callBack page

The directory /<WebLine dir>/pub/html/forms/ on the Collaboration server contains three HTML files that comprise a callback page designed specifically for use with an ACD. You can copy the HTML source from these files into the callback form used for your site.

The table below describes these files:

HTML File	Description
callBackACD.html	Callback button that points to the callback frame.
callFrameACD.html	Frame that contains the callback form.
callFormACD.html	HTML form served to the caller for a callback request.

The file callFormACD.html contains several fields that help you accomplish specific functions with Media Blender. These are:

- Defining a routing address, used for skills-based routing (RoutingAddr)
- Identifying the extension of the Blender agent used by Media Blender to log in to the WebLine Collaboration Server
- Defining call type (optional)
- Gathering caller data (optional--for Aspect switches only)

Define a routing address

The file callFormACD.html contains a field called RoutingAddr. You can set this field to identify where the call should be routed, based on information the caller enters on the form (zip code, account number, and so forth). Use this form as a model when setting up the callback page that should be served to callers.

The Media Blender administrator uses the values set in the RoutingAddr field when setting up a skills table for skills-based routing. See *Setting Up Skills-Based Routing* for more information.

Identify *Blender Agent* used by Media Blender

You identify the Blender Agent used by Media Blender to log into the WCS using the Extension field (in callFormACD.html.) This value is usually 125. Note that this number must be identical to the permanent extension of the Blender agent in the WCS, and referred to in the properties file for the WebLine Collaboration Medium.

In configurations that consist of one WCS and several Media Blenders and phone switches, you must create a Blender Agent on the WCS for each Blender/switch. For example, if the permanent extension of your first Blender agent is 125, you might use 126 as the permanent extension of an additional Blender Agent.

Define Call Type

How you define your call type field in your callBack page depends on whether you are using WCS release 3.0 or WCS release 2.51:

WCS release 3.0:

If you are using Media Blender 3.0 with Collaboration 3.0, you can define a unique calltype variable on the callback page. If you do so, be sure to

point to the correct variable using the calltypefield property in your ACD properties file. (You can also choose to use the Appl1 field already supplied with the sample.)

WCS release 2.51

If you are using Media Blender 3.0 with Collaboration 2.51, you can use the hidden fields Appl1, Appl2, Appl3, and Appl4 to define call type. In the properties file for your ACD, you define which of these fields you want to use to define call type using the calltypefield property (see [Establish call types and outbound dialing strategies](#) for more information.)

All ACD types except Lucent ICM

For all ACD media other than Lucent ICM, the Blender Administrator defines call type codes in a call type table. For example, the file might use the code *von* to identify a voice-over-IP call, or *chat* to identify a chat session. See [Establish call types and outbound dialing strategies](#) for more information.

Defining Lucent ICM call type

For Lucent ICM media, call type codes must be set as follows:

Value in Applx field	Description
voice	Voice-over-IP call
cbncoll	PSTN call or 2 line callback
chatter	Chat session

Gathering Call Data on (Aspect media only)

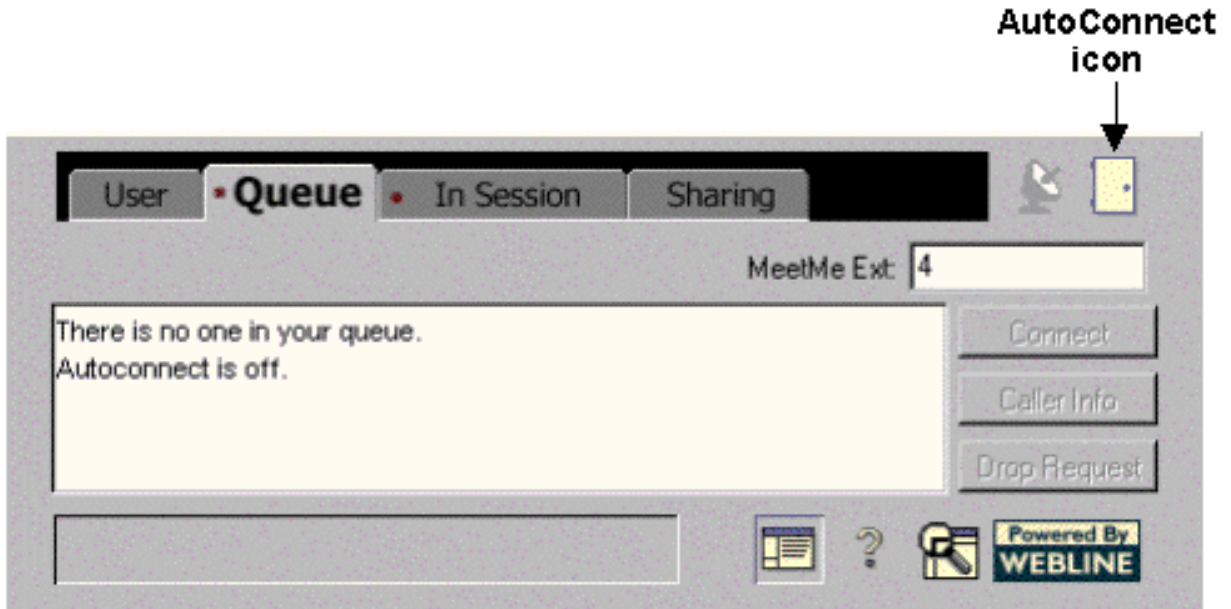
The hidden fields Appl2, Appl3, and Appl4 that appear on callFormACD.html can be set to provide information about the caller to the agent. The strings that appear in these fields are passed into these fields:

Appl1	DataA
Appl2	DataB
Appl3	DataC
Appl4	DataD


These data fields are set into the call requests so that they are available for use in ACD routing logic.

Turn off meetMe AutoConnect

The AutoConnect WCS feature allows meetMe requests to automatically connect to an agent when the agent becomes available. When this feature is turned on, meetMe requests go into session immediately as they are received. It is represented by the AutoConnect icon on the agent UI.



When AutoConnect is off, the icon appears as in the above image--a closed door. When AutoConnect is on, the

icon changes to an open door: 

The AutoConnect icon does NOT affect blended Collaboration requests. You can ensure this icon does NOT appear on the UI of agents who will be receiving blended Collaboration requests (i.e. *callback* requests) by editing the following property in the agent.properties file, located in the <weblinedir>/servlet/properties folder on the WCS.

```
agent.properties  
  
enableAutoConnectIcon=
```

Set this property to false to ensure that the AutoConnect icon, used for meetMe requests, is not displayed on the UI of agents using blended Collaboration.

Note: Keep in mind that, regardless of whether you've removed the icon from the agents UI, individual agents can still activate the AutoConnect feature using user connection and startup preferences. See [Understanding Blended Collaboration](#) for more information.

Set up email notification

Media Blender issues alerts to notify key personnel of potential problems with Media Blender/ACD/WCS connections. Alerts appear on the Media Blender control panel when problems occur. Additionally, you can configure Media Blender to automatically send a list of Blender alerts to a particular email address. That way, you can ensure problems are communicated to individuals responsible for administering Media Blender.

You set up email notification using the Media Blender properties file. Using blender properties, you can enter values used in the email header, such as the recipient of the email notification, the sender of the notification, and text for the subject field.

Note: Keep in mind that, if the email server goes down, Media Blender will be unable to deliver email notification. Also note that any network problems you have may prevent Media Blender from reaching the mail server. Therefore, you should monitor the server using the /status alias in addition to using email notification.

Email notices sent by Media Blender can include descriptions of:

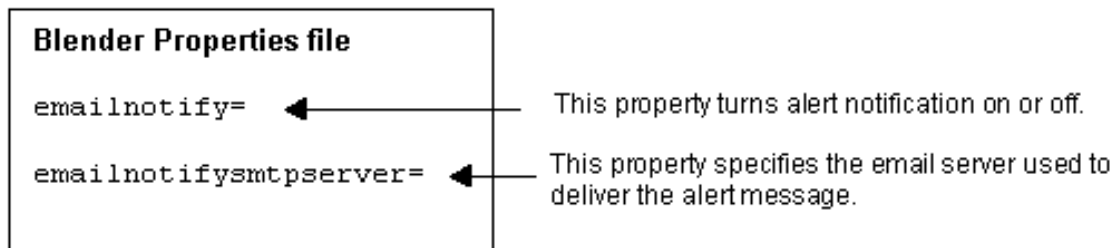
- New alerts--alerts that are appearing for the first time
- Current alerts--alerts that have appeared in previous email notices, but are still active
- Dropped alerts--alerts that have appeared in previous notices and are no longer active (i.e. the problem has been resolved)
- Media Blender messages--messages that report on one-time only Media Blender events. (Examples of Media Blender messages are *Startup complete* and *Shutdown complete*, sent on Blender startup and shutdown.)

Email notification properties

You configure setting for email notification in the properties file for your Media Blender (blender.properties). This section provides a brief overview of some of the Blender properties used to set up Alert Notification. See the Media Blender Reference Guide for a complete listing of alert notification and other Blender properties.

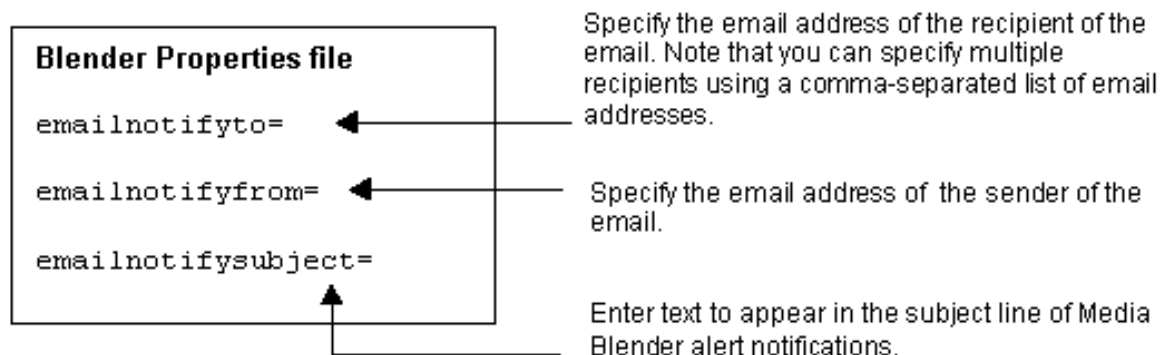
Turning email notification on

Use the following properties to turn email notification on or off, identify the SMTP server, and define information for the email header.



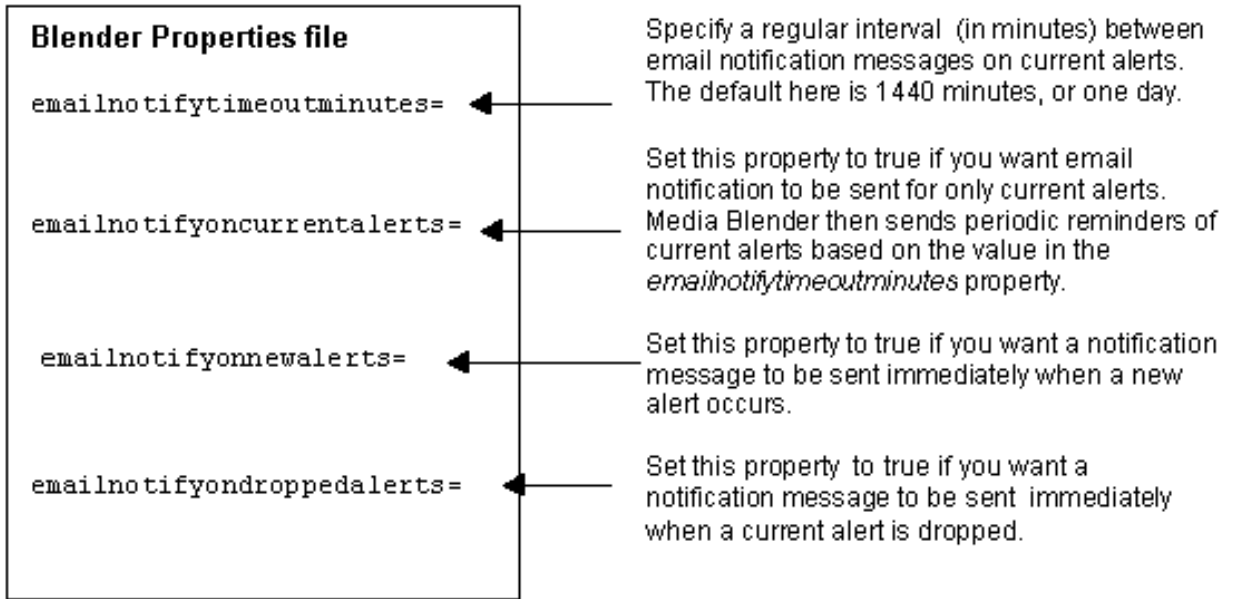
Specifying sender, recipients, and subject

Use these properties to specify the sender, recipient, and subject of Media Blender email notices.



Specifying when email notices are sent

The following properties determine when alert notification messages are sent:



How these properties affect when email notification is sent

In the following example, assume that the `emailnotifytimeoutminutes` value is set at the default: 1440 minutes.

If this property...	...is set to true...	...is set to false...
<code>emailnotifyoncurrentalerts=</code>	Media Blender sends email notifications periodically, once a day, when current alerts exist.	Media Blender does not send regular notification of current alerts. Instead, Media Blender refers to the values set in <code>emailnotifyonnewalerts</code> and <code>emailnotifyondroppedalerts</code> . If these are set to true, Media Blender sends alerts only when new or dropped alerts occur.
<code>emailnotifyonnewalerts=</code>	Media Blender sends notification immediately as a new alert occurs. Note that the notification includes all current and dropped alerts as well as the new alerts.	Media Blender does not send immediate notification as new alerts occur. If <code>emailnotifyondroppedalerts</code> is true, Media Blender sends immediate notification as alerts are dropped. If <code>emailnotifyoncurrentalerts</code> is true, Media Blender sends periodic notification of all alerts based on the value set in the <code>emailnotifytimeoutminutes</code> property.

emailnotifyondroppedalerts=	Media Blender sends notification immediately as an alert is dropped. Note that the notification includes all current and new alerts as well as the dropped alerts.	Media Blender does not send immediate notification as alerts are dropped. If <i>emailnotifyonnewalerts</i> is true, Media Blender sends immediate notification as new alerts occur. If <i>emailnotifyoncurrentalerts</i> is set to true, Media Blender sends periodic notification of all alerts based on the value set in the <i>emailnotifytimeoutminutes</i> property.
-----------------------------	--	---

Understanding emailnotifytimeoutminutes

Be aware that Media Blender applies the value in `emailnotifytimeoutminutes` directly to the most recently received email notification. In other words, the `emailnotifytimeoutminutes` default does **not** ensure that you will receive email notification at the same time each day. Instead, Media Blender sends notification 1440 minutes after the most recently sent notification. Consider this example:

```
emailnotifytimeoutminutes=1440
emailnotifyonnewalerts=true
emailnotifyoncurrentalerts=true
emailnotifyondroppedalerts=true
```

Assume that Media Blender sends a notice at 10:00 a.m. on Monday and a new alert then occurs at 12:15 p.m. Media Blender immediately sends another notice, detailing the new alert and including any current or dropped alerts as well. If no new or dropped alerts occur after that time, Media Blender sends the next notice 1440 minutes after the most recently sent notice--or at 12:15 p.m. on Tuesday.

Including email headers and footers

You can also define a header and footer to be included in email notification messages:

Blender Properties file

```
emailnotifyheader=
emailnotifyfooter=
```

These properties let you specify text files to be appended to the email. The text file you specify in *emailnotifyheader* appears before the alerts; the text file you specify in *emailnotifyfooter* appears after the alert. The example alert notification below contains an `emailnotifyheader`.

For example, you might decide to include a link to the Media Blender control panel in the header or footer file. That way, the recipient can access the control panel quickly, from the email message.

Alert Notification Example

The following sample email alert message contains a current and dropped alert:


```
emailnotifysubject=
↓
Subject: Blender Notices
Date: 10 Sep 1999 12:04:21 -0400
From: blender@mycompany.com ←emailnotifyfrom=
emailnotifyto= → To: joe.smith@mycompany.com

Blender URL: http://blender@mycompany.com/ ←emailnotifyheader=

--- Current Alerts ---

Medium Aspect_ACD: Invalid CCT Error: check CCT #730

--- Dropped Alerts ---

Medium Aspect_ACD: Error on status query - link may be down!
```

For a complete list of Media Blender alerts, see *The Media Blender Reference Guide*.

Using Multiple Media Blenders

Sites that use multiple phone switches can be configured to use multiple Media Blenders with one WebLine Collaboration Server (WCS). Setting up a multiple-blender configuration consists of these basic tasks:

1. Establish Blender agents on the WCS for each switch
2. Configure callback forms to direct requests to the correct Media Blender and switch
3. Identify which WCS agents work on which switch

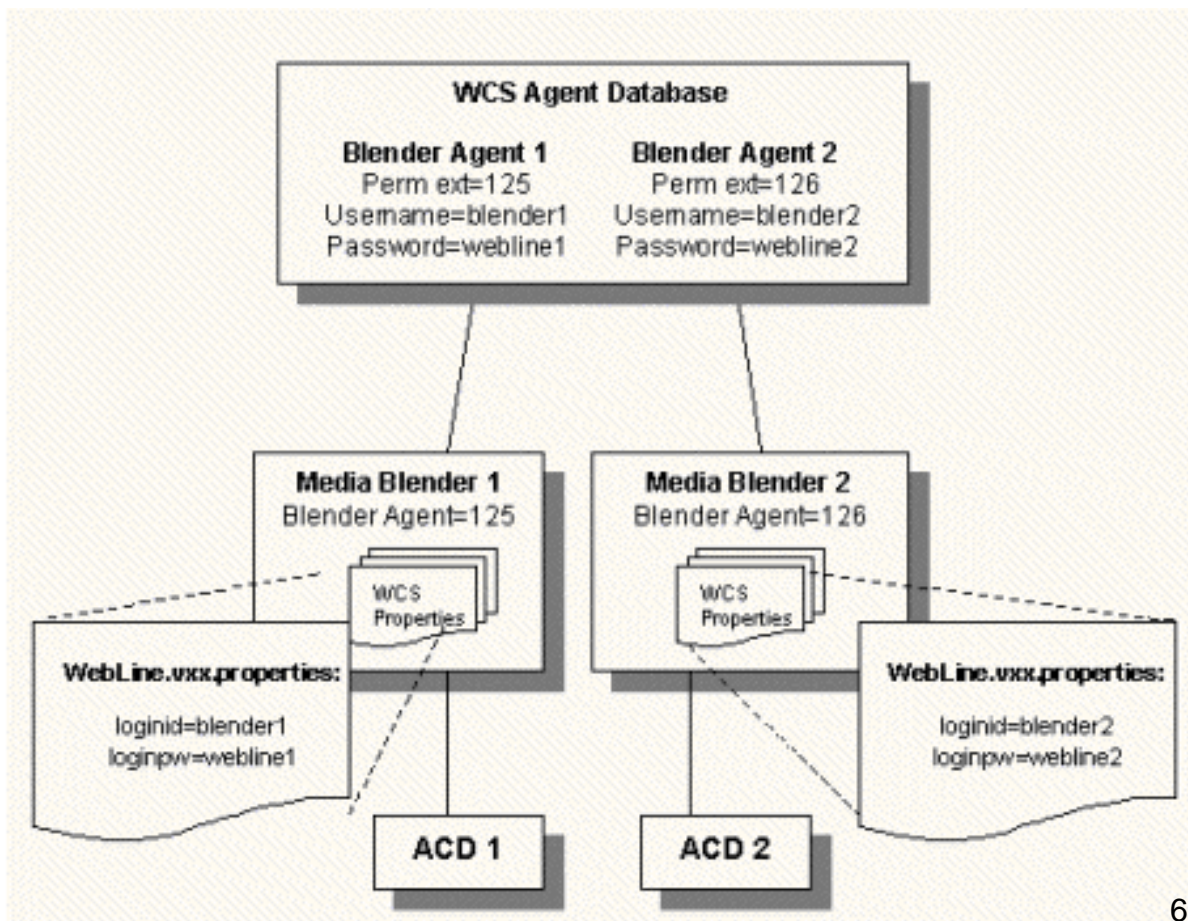
Step 1: Establish Blender Agents on the WCS for each switch

When setting up WCS to work with Media Blender, you must set up a Blender Agent in the agent database of the WCS. Typically, the permanent extension of the Blender agent is 125. When your configuration includes more than one Media Blender, you must establish agents on the WCS for each Blender machine.

Note: Do not confuse the Blender Agent with Collaboration agents who actually sign on to the WCS and switch. See Setting up the Blender Agent in [Configure the Collaboration Server](#) for more information.)

You must also identify the login ID and password for each Blender Agent on each Blender server. A Collaboration properties file resides on each Blender server. This file is typically called `xx.properties` file and resides in the `<weblines dir>servlet/properties` directory. The `loginid` and `loginpw` properties allow you to identify each Blender agent's login ID and password. The values you enter here must match those set up in the agent database on the Collaboration Server.

Consider the following example:



In the example above, you must add two Blender agents to the agent database on the WCS. You might set the permanent extensions for the two Blender agents as 125 and 126. And, on each Blender server, you must add the Blender Agent's login ID and password to the Collaboration properties file.

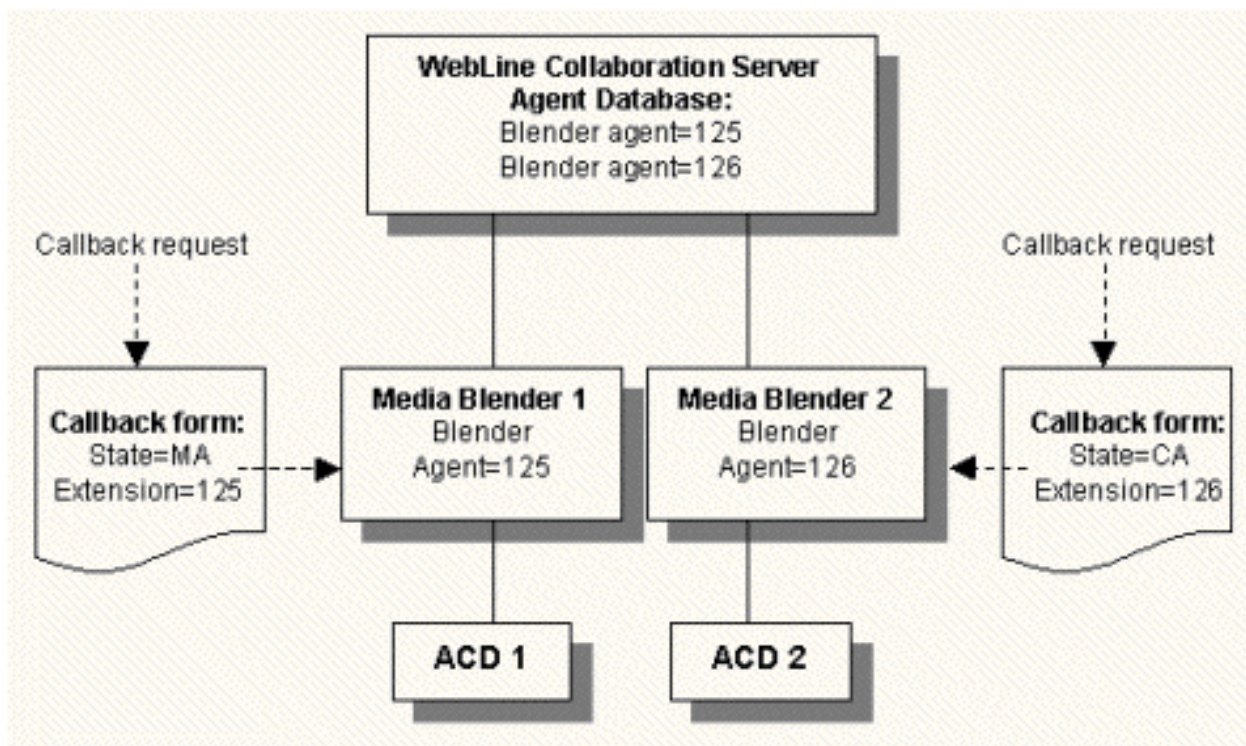
Step 2: Configure callback request forms to direct calls to the correct Media Blender

Collaboration serves callback request pages to callers who request callback. When such requests should be routed through Media Blender, you must identify the Blender agent set up on the WCS by setting the Extension field on the callback HTML form to the Blender agent's permanent extension. In our sample HTML form, callFormACD.html, the Extension field is set to 125. (See [Setting up the Blender Agent](#) for more information.

When your configuration includes more than one Media Blender, you must ensure that all Blender Agents are identified. You can do so using multiple callback forms, serving the appropriate form to callers based on a selection they make when they request callback. For instance, your Web site might instruct callers to click different links to request callback from different groups of agents. Each link would serve a separate callback request form, each identifying a different Blender agent.

Another alternative is to configure your callback HTML form with Javascript that dynamically sets the Extension field to the appropriate Blender agent depending upon some specified user input. This way, you can maintain only one callback form on the WCS.

In the example below, the value in the Extension field is set dynamically, based on what the caller enters in the State field on the callback page.



Step 3: Identify which WCS agents work on which switch

Each Media Blender in your configuration needs to be able to determine which WCS agents work on which switch. By creating an *agent filter file* on each Media Blender machine, you can identify only those agents whose requests will be routed by the Blender's switch.

We provide a sample agent filter file, `agentfilter.properties`, which resides in the `<weblinedir>/servlet/properties` directory. An agent filter file simply lists the logical IDs of all agents who work on the switch configured with the Media Blender. A sample agent filter file appears below:

```
# Agents taking Sales calls
34200 34201 34203 34204 34205 34206 34207 34208 34209 34210

# Agents taking tech support calls
34180 34181 34182 34183 34184 34185 34186 34187 34188
```

In the agent filter file, agent logicalIDs can be separated by spaces or be listed on separate lines. Lines beginning with the pound sign (#) are comments and are ignored.

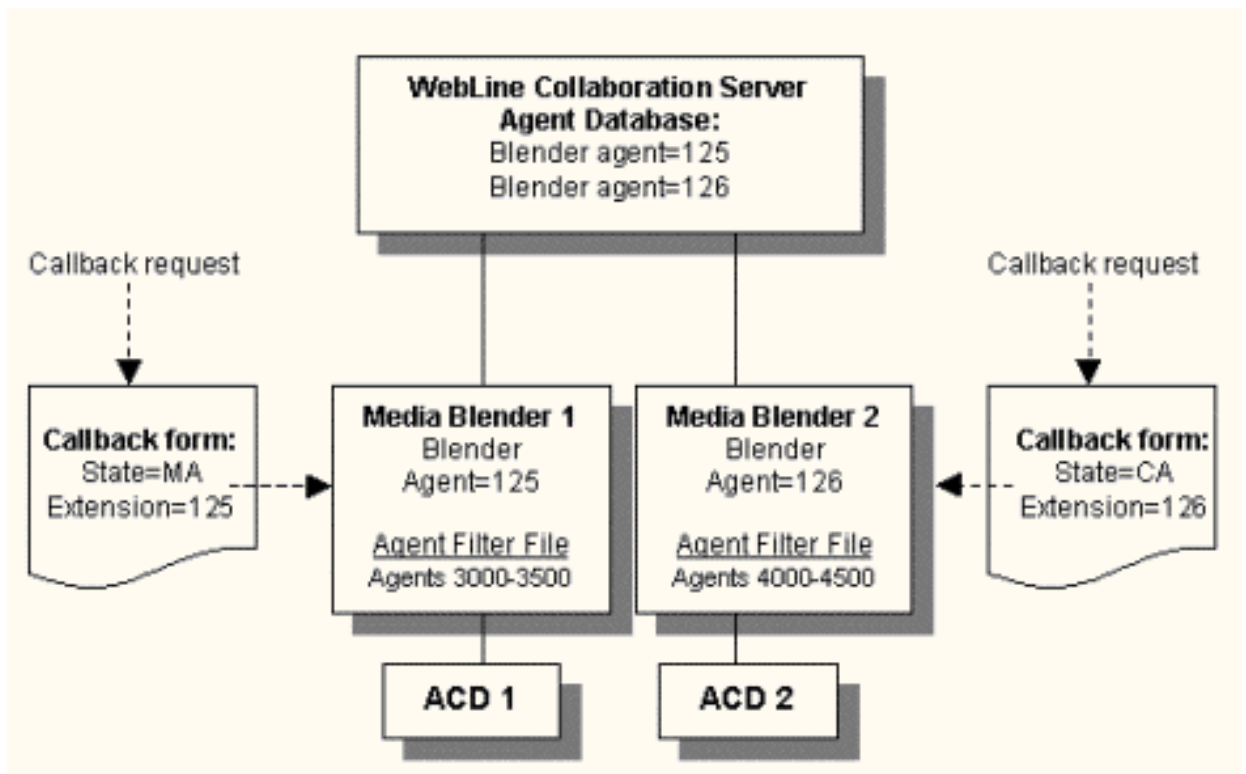
Why filter agents?

When using more than one Media Blender, you must filter agents using an agent filter file. Otherwise, the WCS will share login and logout events with all Media Blenders, and therefore all switches. Therefore, without filtering for every agent log in on the WCS, there will be one successful login on the correct switch and one failed or erroneous login on the incorrect switch.

Identify the agent filter file to all media

Once an agent filter file appears on each Media Blender, the file name must be identified in the properties files for all media configured with each Blender. Use the `shareagenteventsfile` and `acceptagenteventsfile` properties to point to agent filter files set up on each switch. (See [Create an agent filter file](#) for more information.)

The figure below illustrates a multi-blender configuration:



In this example, requests that identify the Blender agent 125 are routed to Media Blender 1 and ACD 1. Once routed there, requests are routed only to WCS agents 3000-3500; WCS agents 4000-4500 are used only with Media Blender 2 and ACD2.

Create an agent filter file

The agent filter file lists the logicalIDs of agents for whom signoff and signon events can be shared and accepted. This file is typically called agentfilter.properties and must reside in the \<weblinedir>\servlet\properties\blender directory.

The agent filter file is used primarily in configurations that include multiple Media Blenders and one Collaboration server. You can also use an agent filter file (which can be used with all media) to establish two groups of agents: those who will use Collaboration with Media Blender and the switch, and those who will use Collaboration only. Agents who do not appear in this file can use Collaboration only.

After you create the agentfilter.properties file, identify it using these properties in your ACD medium's properties file: acceptagenteventsfile and shareagenteventsfile.

Create agentfilter.properties

The agent filter file lists the logical ids of each agent for whom events can be shared and accepted by the switch. A sample agent filter file appears below:

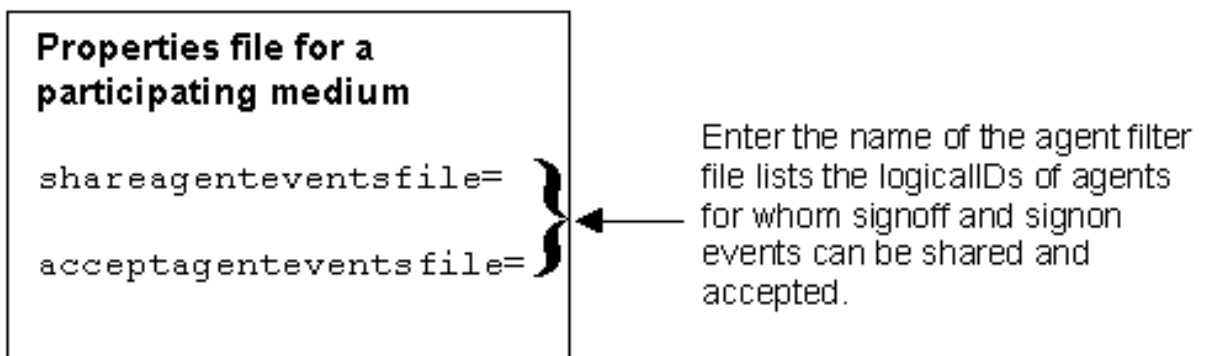
```
# Agents taking Sales calls
34200 34201 34203 34204 34205 34206 34207 34208 34209 34210

# Agents taking tech support calls
34180 34181 34182 34183 34184 34185 34186 34187 34188
```

You can separate agent IDs either by using spaces or by listing them on separate lines. Lines beginning with a pound sign (#) are comments and are ignored.

Identify the agent filter file to the participating media

Once the agent filter file has been created, use these properties to identify it in the properties files for all participating media:



Appendix A: Terminology

agent

An individual who receives and handles customer calls and web-based requests within a call center.

blended Collaboration session

A Collaboration session blended with the ACD by Media. Blended Collaboration sessions typically begin when a caller submits a web-based request by clicking a callback button on a web page. The caller completes a callback form and Collaboration retrieves caller information (i.e. name, phone number, skill group, etc.) Media Blender then blends the information over to the ACD, which provides call back to the customer.

blender agent

An agent, set up on the Collaboration server, that enables Media Blender to log in to Collaboration. Each Media Blender has one blender agent on the WCS.

Callback button

A button placed on a web site used by the caller to initiate a blended Collaboration session.

Callback page

A form served to the caller to retrieve caller information, such as name, telephone number, and skill group.

Caller

An individual submitting a phone call or web-based request to a call center.

CTI driver

Software designed to accommodate the CTI package/middleware used in a Media Blender configuration.

CTI Strategy

Software that determines the callflow of the outbound call to the caller.

Media Blender Administrator

An individual responsible for installing, configuring, and administering Media Blender.

medium

An electronic form of session-based information. Media Blender shares CTI events between participating media. In a typical installation, Media Blender shares events between a web medium (the WCS) and an ACD medium (the switch)

meetme session

On the Collaboration server, a session initiated by a caller entering an agent's permanent extension. Meetme sessions are not blended by Media Blender; they typically begin with a phone call from the caller to the agent. The caller obtains the agent's permanent extension and specifies it when connecting to the call center via Collaboration.

phantom line

Phone lines set aside for providing callback to customers. Used with Phantom line CTI strategies, phantom lines wait in queue on behalf of the caller , ensuring the caller receives callback only when an agent is available.

routing logic

Logic set up on the ACD to ensure calls are routed to agents who possess appropriate skills.

switch

An Automatic Call Distributor (ACD) or PBX.

Switch Administrator

An individual responsible for ACD administration. The switch administrator must work with the Media Blender Administrator to ensure proper communication between Media Blender and the ACD.

Web Administrator

An individual responsible for creating and mainting HTML pages and forms as they relate to Media Blender.

WebLine Media Blender Version 3.0

© 1999 by Weblin Communications Corporation. WebLine, Media Blender, ScriptShare, FormShare, and Follow Me Browsing are trademarks of WebLine Communications Corporation. Other products and brand names are trademarks of their respective owners.