SWYX
COMMUNICATIONS AG

CDR Specification SwyxServer v4.10

Swyx Communications AG
Joseph-von-Fraunhofer-Str. 13a
44227 Dortmund

Status: Reviewed version,

Number of pages:	14		
Version:	2		
Created:	15.01.03	by:	Martin Hüser
Last modification:	17.01.03 12:51	by:	Uwe Sauerbrey
File:	N:\documentation\SwyxWare\v4.1\CDR Specification SwyxServer v4.10.doc		

1.1 Call details and active call display

Changing and extending call details records is necessary for several reasons. First the information stored in a CDR is not sufficient to log call transfers completely. CDRs in SwyxWare v4.0 allow assigning the call costs to the correct originator, but lack information about the resulting call of a transfer. For some call-center scenarios much more information is needed, e.g. to allow tracking how long a call is being alerted. The conference establishing mechanism via call transfers lead to a problem with CDRs and the active call display within SwyxWare administration, too. Let User C have a call to A and B:

C-> A
B -> C

M establishes a conference which is done by establishing a call to the ConferenceManager Mfor itself and two other calls to the conference manager M with immediate call transfer of the calls from/to A and B. This leads to the following CDRs:

C => A	(transferred)
B => C	(transferred)
C => M	(connected)
C => M	(transferred)
C => M	(transferred)

These CDRs are written into the CDR file after the calls are finished. This is somewhat confusing, but allows assigning the call costs to the client who causes them. But with the v4.0 CDR format it's not possible to detect which calls are part of a conference. Therefore CDRs will include information to see which CDRs are related, e.g. from which call another was transferred. SwyxWare v4.10 will extend call detail records and change CDR handling during transfer as follows:

1.1.1 CDR file format

The new CDR file format is as follows (new/changed columns in *italics*)

Name	Description
CallId	*Unique call ID. Each call (or CDR) gets an ID.*
OriginationNumber	Call originator's number. For internal calls this is the number the client has specified during call establishment. For external calls via gateway this is the number signaled from PSTN. For external calls via a link this is the originator's number in complete form (+492314777222)
OriginationName	Call originator's name. SwyxServer tries to get the name corresponding to the number by searching: The client defined name, the SwyxWareWare user name, or the SwyxWare phonebook name
CalledNumber	Called party number. This is the orignally dialed number. If a User calls number "103" this field will contain "103" even if the call is redirected or picked up by someone else.
CalledName	Called party name. SwyxServer tries to determine the name in the same way as for "OriginationName"
DestinationNumber	Number of client who accepted the call. For non-picked-up and non-redirected calls this is the same as called number. Otherwise it defines who accepted the call.
DestinationName	Name corresponding to destination name. . SwyxServer tries to determine the name in the same way as for "OriginationName"
StartDate	Start date of the call. This defines the time when SwyxServer received the

	call from the client
StartTime	Start time of the call. See above.
ScriptConnectDate	*Date when a call routing script accepted the call. For incoming calls only*
ScriptConnectTime	*Time when a call routing script accepted the call. For incoming calls only*
DeliveredDate	*Date when a call routing script delivered the call, i.e. when the ConnectTo() function is called. For incoming calls only*[1]*.*
DeliveredTime	*Time when a call routing script delivered the call, i.e. when the ConnectTo() function is called. For incoming calls only*[2]*.*
ConnectDate	*Date when the called destination accepted the call.*
ConnectTime	*Time when the called destination accepted the call.*
EndDate	End date of the call. This is the time when the call was terminated
EndTime	End time of the call. This is the time when the call was terminated
Currency	Currency of charges. If AOC is "1" and PSTN has signaled charging information with currency, this column contains this currency. If AOC is "1" and PSTN hasn't signaled charging information this value is the charging unit currency configured within SwyxWare Administration. If AOC is "0" this is the currency of the Least cost routing module.
Costs	Call costs. If AOC is "1" and PSTN has signaled complete charging information in currency units, this column contains these signaled charges. If "AOC is "1" and PSTN has signaled charging units only, this column contains the calculated charges as configured within SwyxWare Administration The value's decimal separator is the SwyxWare service accounts default decimal separator.
State	Call state: Alerting: Call was terminated whilst being alerted at the destination Connected: Call was terminated whilst being connected at the destination OnHold: Call was terminated whilst begin on hold Transferred: Call was terminated whilst being connected after a call transferr
PublicAccessPrefix	The public access prefix dialed, if any.
LCRProvider	The Call-by-Call number dialed, if any
ProjectNumber	The project number dialed, if any
AOC	"1" advice of charge received from PSTN "0" costs are calculated by the Least cost routing module
OriginationDevice	Device which initiated the call. This is the SwyxGate name for incoming PSDN calls, the link name if call was routed via a link and empty otherwise. Note that SwyxWare Administration may show "(call routing)" during a call, when a callrouting script executes a ConnectTo() function-
DestinationDevice	Destination device where the call is targeted at. For calls into PSTN this is the SwyxGate used. For external calls to another site this is the name of the link used to route the call. Contains "(call routing)" if the call is accepted by a user's call routing script, e.g. to play or record a message.
TransferredByNumber	*Number of user who transferred the call. Empty if call not participated in a call transfer*
TransferredByName	*Name of user who transferred the call. Empty if call not participated in a call transfer*
TransferredCallId1	*CallId of the first participating call in case of a call transfer*
TransferredCallId2	*CallId of the second participating call in case of a call transfer*
TransferredToCallId	*Describes that this call has been transferred to the specified callId.*
TransferDate	*Date of call transfer. Valid for call which have been transferred.*
TransferTime	*Time of call transfer. Valid for call which have been transferred.*

[1] Every call has a call routing script. If no user defined one, the server default script is used. Therefore this value is set for every incoming call.

DisconnectReason

Reason for call disconnection. A string describing the disconnect reason. The following values are possible:

Reason	***Description***
Busy	*Destination busy*
Reject	*Destination rejected the call*
NoAnswer	*Destination did not answer*
TooLate	*Another (not the called) device accepted the call*
UnknownNumber	*An unknown number was called*
Unreachable	*Destination unreachable*
NetworkCongestion	*PSTN network congestion*
BadFormatAddress	*Address format incorrect*
CallRoutingFailed	*Call routing failed, e.g. the call routing script could not be started..*
PermissionDenied	*No sufficient permission for the call*
CallDisconnected	*Normal call disconnect*
IncompatibleDestination	*Originator and destination not compatible, e.g. no matching voice codecs*
NumberChanged	*Destination number in PSTN has changed*
NoChannelAvailable	*No SwyxWare channel available*
OriginatorDisconnected	*Call originator disconnected the call*
CallTransferred	*Call has been transferred, TransferredToCallId links to the resulting CDR, TransferDate/Time contains the time when the transfer was initiated .*

SwyxServer handles Call Details records by means of CDR objects which are created, updated, written and deleted during a call. CDR handling for different call types is explained in detail in the following sections. These sections show several CDRs as tables. Empty CDR column may be omitted for brevity. CDR columns which are not relevant for the specific scenario may be omitted as well. Each CDR file will have a version number on the first line. Column headers follow on the second line.

1.1.2 CDR for simple calls

If a user A establishes a call to User B a CDR object is generated at the beginning of the call which contains the following columns:

CallID	3
OriginationNumber	“123”
OriginationName	“User A”
StartDate	“19.11.2002”
StartTime	“13:03:23”
State	“Initialized”

If SwyxServer has resolved the dialed number and is alerting the call, the CDR object is updated to:

CallID	3
OriginationNumber	“123”
OriginationName	“User A”
CalledNumber	“456”
CalledName	“User B”
StartDate	“19.11.2002”
StartTime	“13:03:23”
DeliveredDate	“19.11.2002”
DeliveredTime	“13:03:24”

State	“Alerting”
ProjectNumber	“4711” (if projectnumber *4711# was dialed)

If B accepts the call, the CDR is updated again to:

CallID	3
OriginationNumber	“123”
OriginationName	“User A”
CalledNumber	“456”
CalledName	“User B”
StartDate	“19.11.2002”
StartTime	“13:03:23”
DeliveredDate	“19.11.2002”
DeliveredTime	“13:03:24”
ConnectedDate	“19.11.2002”
ConnectedTime	“13:03:28”
State	“Connected”
ProjectNumber	“4711” (if projectnumber *4711# was dialed)

After A and B have finished their call, the CDR is updated again and written into the CDR file.

CallID	3
OriginationNumber	“123”
OriginationName	“User A”
CalledNumber	“456”
CalledName	“User B”
StartDate	“19.11.2002”
StartTime	“13:03:28”
DeliveredDate	“19.11.2002”
DeliveredTime	“13:03:24”
ConnectedDate	“19.11.2002”
ConnectedTime	“13:03:28”
EndDate	“19.11.2002”
EndTime	“13:05:48”
State	“Connected”
ProjectNumber	“4711” (if projectnumber *4711# was dialed)

1.1.3 CDR for external calls

For external calls some additional information is written into the CDR objects. If User A calls an external party and SwyxServer uses “SwyxGate” with CAPI Controller 1 to route the call, the following additional information is filled in the CDR:

PublicAccessPrefix	“0”
LCRProvider	“01033” (if LCR is enabled and LCR provider found and used)
DestinationDevice	“SwyxGate|1”

For incoming external calls, the “OriginationDevice” is filled with the gateway/controller used for this call. If charging information is available the charging information if filled in accordingly. See section 1.2 below for details.

1.1.4 CDRs when executing call routing scripts

When executing a call routing scripts more than one CDRs are active for a call. Consider the following example: User B has a call routing script which accepts the call, plays a message and tries to deliver the call to his internal client first. If that fails, the script tries to route the call to his mobile phone. If that isn't successful, too, the caller may leave a message on the user's voicebox. User A calls this user.

1. A CDR object is generated for the incoming call from User A:

CallId	3
OriginationNumber	"123"
OriginationName	"User A"
StartDate	"19.11.2002"
StartTime	"13:03:23"
State	"Initialized"

2. Address resolution finds the destination, "User B", SwyxServer starts the script. As soon as the script reaches the "PlayMessage" instruction, the call is alerted for a couple of seconds and connected after that. The CDR looks like this:

CallId	3
OriginationNumber	"123"
OriginationName	"User A"
CalledNumber	"456"
CalledName	"User B"
StartDate	"19.11.2002"
StartTime	"13:03:28"
ScriptConnectDate	"19.11.2002"
ScriptConnectTime	"13:03:30"
State	"Connected"
DestinationDevice	"(call routing)"

This CDR shows that the call from User A with number 123 to User B's number 456 is "connected" to the call routing script, not to User B itself.

3. After playing the message, the script reaches a "Connect to original destination" instruction. SwyxServer set the "DeliveredDate" and "DeliveredTime" for CallId 3:

CallId	3
OriginationNumber	"123"
OriginationName	"User A"
CalledNumber	"456"
CalledName	"User B"
StartDate	"19.11.2002"
StartTime	"13:03:28"
ScriptConnectDate	"19.11.2002"
ScriptConnectTime	"13:03:30"
DeliveredDate	"19.11.2002"
DeliveredTime	"13:03:35"
State	"Connected"
DestinationDevice	"(call routing)"

For the call to the destination B SwyxServer generates a new call with the script as “originator” and User B as called party. This new call has the following CDR object:

CallId	3
OriginationNumber	“123”
OriginationName	“User A”
StartDate	“19.11.2002”
StartTime	“13:03:35”
State	“Initialized”
OriginationDevice	“(call routing)”

Note that this new CDR gets the same CallId as the original call. This call id is part of the incoming call event sent to the destination client. If this call is alerted, CDR state changes to “Alerting” and the called number/name is set to 456/User B. User A is still connected to the script. If User B doesn’t accept the call, the script continues, the above CDR object is deleted from memory. Such CDRs can be viewed in the active call list in SwyxWare administration, only.

4. The script continues and tries to deliver the call to mobile phone number “0016012345678”. The DeliveredDate and DeliveredTime is updated for CallId 3, a new call is generated, “User A” remains connected to the script and a new CDR object is created. If the call is alerted at the destination the CDR looks like this:

CallId	3
OriginationNumber	“123”
OriginationName	“User A”
CalledNumber	“0016012345678”
CalledName	“”
StartDate	“19.11.2002”
StartTime	“13:03:55”
State	“Alerted”
OriginationDevice	“(call routing)”
DestinationDevice	“SwyxGate|1”

5. The call is accepted at the mobile phone. SwyxServer transfers the destination call to User A which disconnects User A from the script. The CDR from step 2 is deleted and the CDR from step 4 is updated and looks like this:

CallId	3
OriginationNumber	“123”
OriginationName	“User A”
CalledNumber	“456”
CalledName	“User B”
DestinationNumber	“0016012345678”
DestinationName	“”
StartDate	“19.11.2002”
StartTime	“13:03:28”
State	“Connected”
ScriptConnectDate	“19.11.2002”
ScriptConnectTime	“13:03:30”
DeliveredDate	“19.11.2002”

DeliveredTime	“13:03:55”
ConnectDate	“19.11.2002”
ConnectTime	“13:03:59”
OriginationDevice	“”
DestinationDevice	“SwyxGate|1”

1.1.5 CDRs during call transfer

CDR handling during call transfer has changed since v4.0. This section describes how it works in v4.10. User C has a call on hold to User A and another call to User B which is currently active, i.e. C and B are speaking together. C has established both calls, i.e. C is charged for the call costs. There exist two CDR objects as follows:

CDR 1

CallID	3
OriginationNumber	“101”
OriginationName	“User C”
CalledNumber	“0016012345678”
CalledName	“User A”
StartTime	“13:08:24”
ConnectedTime	“13:08:45”
State	“On Hold”
OriginationDevice	“”
DestinationDevice	“SwyxGate|1”

CDR 2

CallID	7
OriginationNumber	“101”
OriginationName	“User C”
CalledNumber	“002314777222”
CalledName	“User B”
StartTime	“13:09:34”
ConnectedTime	“13:09:56”
State	“Connected”
OriginationDevice	“”
DestinationDevice	“SwyxGate|1”

If User C initiates a call transfer, C is disconnected and “User A” and “User B” are speaking. Both CDRs are updated and look like this:

CDR 1

CallId	3
OriginationNumber	“101”
OriginationName	“User C”
CalledNumber	“0016012345678”
CalledName	“User A”
StartTime	“13:08:24”
ConnectedTime	“13:08:45”
State	“Transferred”
OriginationDevice	“”
DestinationDevice	“SwyxGate|1”
TransferredToCallId	8
TransferDate	“19.11.02”
TransferTime	“13:10:06”

CDR 2

CallId	7
OriginationNumber	“101”
OriginationName	“User C”
CalledNumber	“002314777222”
CalledName	“User B”
StartTime	“13:09:34”
ConnectedTime	“13:09:56”
State	“Transferred”
OriginationDevice	“”
DestinationDevice	“SwyxGate|1”
TransferredToCallId	8
TransferDate	“19.11.02”
TransferTime	“13:10:06”

A new CDR is created for the “new” call from User A to User B:

CDR3

CallId	8
OriginationNumber	“0016012345678”
OriginationName	“User A”
CalledNumber	“002314777222”
CalledName	“User B”
StartTime	“13:10:06”

ConnectedTime	“13:10:07”
State	“Connected”
OriginationDevice	“SwyxGate|1”
DestinationDevice	“SwyxGate|1”
TransferredByNumber	“101”
TransferredByName	“User C”
TransferredCallId1	3
TransferredCallId2	7

As long as A and B speak, all three CDR objects remain in memory. Charging information may be received from A or B and are counted in CDR 1 and CDR 2. If the call is disconnected all three CDRs are written into the CDRs file after setting the call end time:

CDR 1

CallId	3
OriginationNumber	“101”
OriginationName	“User C”
CalledNumber	“0016012345678”
CalledName	“User A”
StartTime	“13:08:24”
ConnectedTime	“13:08:45”
EndTime	“13:15:44”
State	“Transferred”
OriginationDevice	“”
DestinationDevice	“SwyxGate|1”
TransferredToCallId	8
TransferDate	“19.11.02”
TransferTime	“13:10:06”
Currency	“EUR”
Charges	“1.23”
AOC	“1”

CDR 2

CallId	7
OriginationNumber	“101”
OriginationName	“User C”
CalledNumber	“002314777222”
CalledName	“User B”
StartTime	“13:09:34”
ConnectedTime	“13:09:56”
EndTime	“13:15:44”
State	“Transferred”
OriginationDevice	“”
DestinationDevice	“SwyxGate|1”
TransferredToCallId	8
TransferDate	“19.11.02”
TransferTime	“13:10:06”
Currency	“EUR”
Charges	“4.33”
AOC	“1”

CDR 3

CallId	8
OriginationNumber	“0016012345678”
OriginationName	“User A”
CalledNumber	“002314777222”
CalledName	“User B”
StartTime	“13:10:06”
ConnectedTime	“13:10:07”
EndTime	“13:15:44”
State	“Connected”
OriginationDevice	“SwyxGate|1”
DestinationDevice	“SwyxGate|1”
TransferredBy	“User C”
TransferredCallId1	3
TransferredCallId2	7

It’s necessary to keep all three CDR objects for the complete call duration to be able to count call costs correctly. Having only CDR3 after the transfer would lead to several problems: Call costs may stem from both sides and each charging event contains the cumulative costs for the call so far. In countries without charging information from PSTN and no LCR support, charges have to be calculated by analyzing the call duration and destination. This would be impossible with CDR3 only.

1.1.6 CDRs when transferring a transferred call

The example from the last section had three CDRs involved:

3: C => A (transferred)
7: C => B (transferred)
8: A => B (connected, transferred from C)

Now let’s assume that B is not an external party but a SwyxWare user. B gets another call from D, accepts it which puts A => B on hold:

10: D => B (connected)

B initiates a transfer to connect D with A. Therefore the following CDRs remain:

3: C => A (transferred)
7: C => B (transferred)
8: A => B (connected, transferred from C)
10: D => B (connected)

To complete the call transfer SwyxServer will update the CDRs and write and delete two of them. Transferred CDRs which have the transferrer B involved (CDR 7 in this case) will be updated to contain the transfer time as end time and will be commited into the CDR file. CDRs in "connected" state but which have been generated during a previous transfer will be updated, written and deleted, too (CDR 8 in this case). As a last step the CDR 10 changes its state to "transferred" and a new CDR 11: D => A is generated:

3: C => A (transferred)
10: D => B (transferred)
11: D => A (connected, transferred from B)

This mechanism keeps three active CDRs when doing call transfers and allows collecting charging information for the correct user. In countries without charging information from PSTN or LCR call costs have to be calculated offline by analyzing the CDR file. The SwyxWare v4.10 CDR scheme makes this quite easy. Each CDR in state transferred or in state connected (without being transferred) allows calculating the call duration from ConnectTime and EndTime. The originator is the person to charge.

1.1.7 CDRs and active call display

Having three CDRs after a call transfer is useful for call analysis and charging calculation but is somewhat confusing for the administrator when displayed in SwyxWare Administration. This is especially true for conferences. A 3-participant conference would produce 5 CDRs. Therefore the active call display will merge CDRs with state "Transferred" to the resulting CDR of the transfer. Consider the example from the previous section:

3: C => A (transferred)
7: C => B (transferred)
8: A => B (connected, transferred from C, and transferred callids: 3, 7)

SwyxWare will deliver only CDR 8 for active call display to SwyxWare Administration. 3 and 7 will be skipped, but their charges will be added and shown in 8. The CDR file contains all 3 CDRs.

1.2 Handling of charging information

SwyxServer's charging information handling has not changed between v4.0 and v4.10 but was never documented. This is done in this section. (The text is a copy of Knowledgebase-Article kb2211)

SwyxWare recognizes charging information sent via ISDN or can calculate them itself via the least cost routing module. This article describes in detail how SwyxWare processes charges, how charges are signalled into Sub-PBXes or ISDN devices connected to SwyxWare. It explains how charging information is forwarded via SwyxLinks and how charging information is stored in call detail records.

There are two sources of charging information

a. Information received on ISDN D-channel via "Advice of charge" information elements (IE).

b. Calculated from the least cost routing module (LCR).

ISDN provided charging information has priority over LCR calculated values always. SwyxServer calculates charging information via LCR periodically as long as no charges are received from ISDN, starting one second after a call has been connected. When SwyxServer gets an AOC IE on ISDN D-channel, LCR charging calculation will be switched off for the call. Usually the first AOC from ISDN is sent immediately after the call is connected so that no LCR calculation will be done at all.

Charging information via ISDN

If SwyxWare receives charging information via ISDN, this information will be used always. No calculation via LCR will be done. Charging information from ISDN can be in two formats. Both formats are available during a call (AOC-D) and/or after the end of a call (AOC-E) and/or at the beginning of a call (AOC-S). SwyxWare supports AOC-D and AOC-E only. Swyx knows no provider who supports AOC-S. It depends on your provider and your line what kind of information will be sent. Some provider are sending charging information as display information elements (Display-IE). The format can vary. SwyxGate can interpret such Display-IEs, too, and allows some adaption via the following registry value:

Location: HKEY_USERS\<SID of the user account the service is running with>\Software\<manufacturer>\IpPbxGate\CurrentVersion \Options\Ctrl<x> where <x> is the CAPI controller number of the line. To get the SID of the SwyxGate service account use the PsGetSid tool (see link below).

Type: REG_SZ

Name: ChargingFormat

Value: The currency value, e.g. “Fr.”. If a display information element begins with this string, SwyxGate assumes the charging value to follow this string.

To get detailed information about the charging information format, contact your ISDN line provider.

Advice of charge in charging units
This type of charging information contains a number of charging units. It's the type of AOC the German Telekom usually sends. To get a display on clients or in call details records as currency value, SwyxServer needs to know how much one charging unit costs. This can be configured within SwyxWare Administration. Open SwyxServer properties and select the "Charges" property page to set the charging unit value. Changing this value has no influence on past calls. They will be use for any call established after changing these settings. **Note that these settings are used for this type of AOC only.** They're ignored for AOC in currency units or LCR calculated values.

Advice of charge in currency units
This type of charging information contains the number of charging units, the value of one unit and the currency. SwyxServer uses these information to generate the charging display for the clients and call details records. E.g. Swiss Telecom usually sends this type of AOC.

Charging information calculated via LCR

If no charging information is available on the ISDN D-Channel (AOC), SwyxServer uses the least cost routing module to calculate charges itself. Least cost routing works in Germany only, i.e. if your SwyxServer is located in Germany and has a german area code. Same is true for charges calculated via LCR. The least cost routing module uses the preselected provider configured within the SwyxServer properties in SwyxWare Administration to calculate charges. Note that charging calculation is independent of the least cost routing itself, i.e. even if you've switched off LCR, call charges will be calculated and displayed. To prevent this, remove the LCR module if it's already installed or deselect it during installation. To remove it, stop all Swyx Services, open Windows 2000 control panel, select the "Add/Remove programs" applet, and click "Change" at the SwyxWare entry. SwyxWare installation starts in maintenance mode. Select the "Modify" radio button and click "Next". Now you can deselect "Least Cost Routing". Finish the installation wizard by clicking "Next". To prevent it from being installed in the first place, deselect "Least Cost Routing" during installation.**Note that if SwyxServer receives charging information from ISDN, LCR calculation won't be used. ISDN provided charges have priority over LCR calculated values always.**

Accuracy

The accuracy of charging information depends on your provider (in case of ISDN provided information) or on the LCR module (when SwyxServer has no information from ISDN and calculates them itself). The LCR module calculates charging information using the configured default rate. SwyxServer default is the german Telekom's common ISDN rate. If you've any special rate for your ISDN line which is not supported by the LCR module, SwyxWare cannot calculate correct values. You can configure the provider/rate with SwyxWare administration. Open SwyxServer properties and select the least cost routing tab. Change the "Preselected Provider" to match the provider/rate of your ISDN line. If the rate of your line and the SwyxWare LCR configuration matches, the calculated charges are quite accurate, but Swyx cannot guarantee that they're identically to the invoice you get from your provider. Furthermore, usually no provider will guarantee that charging information signalled via ISDN is 100% accurate, too.

Charging information in Sub-PBXes or ISDN devices

Beginning with SwyxWare v4.0 charging information will be signalled to another PBX connected to Swyxware as Sub-PBX or an ISDN device, i.e. with the ISDN board operating in NT-mode and SwyxWare line configuration set to "ISDN line to sub-PBX or ISDN device". SwyxWare sends AOC-D information elements in currency units on the D-channel always, independent from the received AOC on PSTN lines. Even charging information calculated by SwyxWare itself via LCR will be send in this format to the sub-pbx or ISDN device.

For PBXes or ISDN devices which cannot handle AOC-D IEs, SwyxGate can be configured to send "display" information elements instead (Display IE). To do this, add the following registry key on the computer where SwyxGate is running:

Location: HKEY_USERS\<SID of the user account the service is running with>\Software\<manufacturer>\IpPbxGate\CurrentVersion \Options\Ctrl<x> where <x> is the CAPI controller number of the line where the Sub-PBX or ISDN device is connected to. To get the SID of the SwyxGate service account use the PsGetSid tool (see link below).

Type: REG_SZ

Name: DisplayFormat

Value: A printf formatting statement, such as "%s %1.2f", which results in something like "EUR 1, 23". %s is replaced by the currency; %1.2f displays the value with one or more digits before the decimal separator and exactly two digits afterwards. Note that you cannot change the order. %s has to be before %f always.

If this value is set, charging information will be signalled on this line as "Display" IE always. If this value is empty or not present (default) charging information will be signalled on this line as AOC-D IE.

Charging information via SwyxLinks

SwyxServer will forward charging information to other sites via a SwyxLink, too. If a client A1 on site A dials a number which is routed via a SwyxLink to site B and a SwyxGate GB on site B into B’s PSTN, the charging information received from SwyxGate GB will be forwarded over the SwyxLink back to client A1. If SwyxGate B doesn’t get charging information via ISDN, SwyxServer on site B calculates charges via LCR and sends them back via the SwyxLink to site A for display at A1 and recording in call detail records.

Charging information in CDRs

Charging information will be recorded in call detail records, too. The following fields contain the information

Currency
Currency of charges. If field AOC is "1" and PSTN has signaled charging information with currency, this column contains that currency. If field AOC is "1" and PSTN hasn’t signaled charging information this value is the charging unit currency configured within SwyxWare Administration.
If field AOC is "0" this is the currency calculated by the Least cost routing module.

Costs
Call costs. If field AOC is "1" and PSTN has signaled complete charging information in currency units, this column contains these signaled charges. If field "AOC is "1" and PSTN has signaled charging units only, this column contains the calculated charges as configured within SwyxWare Administration. Costs are displayed in 1/100th of the currency, e.g. cents for Euro.

AOC
"1" advice of charge received from PSTN or another site.
"0" costs are calculated by the Least cost routing module.

DestinationDevice
Destination device where the call is targeted at. For calls into PSTN this is the SwyxGate used. For external calls to another site this is the name of the link used to route the call. Contains "(call routing)" if the call is accepted by a user’s call routing script, e.g. to play or record a message. You can use this field to determine if the recorded charges are from your site (SwyxGate name) or from another site (link name)