



Blaise Services At Westat Technical Notes Publication

Technical Note on Daybatch Management

Copyright © 2002 Westat. Not to be reproduced without written permission.

While precautions have been taken in preparing this Technical Note, Westat assumes no responsibility for errors, omissions, or damages resulting from the information contained herein. Westat makes this information available to you "as is" and without warranty of any kind, either express, implied or statutory.

Technical Notes are intended to facilitate the sharing of useful technical information about issues and solutions concerning the use of Blaise. These notes arise from our Blaise work and we believe they may be of use in other situations. The ideas contained in this Technical Note may or may not be appropriate for your specific needs. We invite users to submit (to blaise@westat.com) relevant issues, resolutions, or questions related to this Technical Note.

This technical note provides an overview of the Blaise scheduler from a programming point of view. It assumes that as one of its readers you are already somewhat familiar with the materials provided in chapters 10 and 11 of the Blaise Developer's Guide.

Specifically, this technical note explains how Blaise goes about initializing and then changing values in fields in the daybatch file and how Blaise uses these values to determine which case should be delivered whenever an interviewer requests a case. Fields in the daybatch file appear in bold typeface. Status categories are italicized. It is worth noting that Blaise uses the status category of *default* to indicate that a case is active.

The Daybatch File

In order to understand Blaise CATI in depth, you must first understand how Blaise initializes, updates and uses the daybatch file. The daybatch file contains the following 14 fields:

- **JoinID** – The unique integer that links a record in the daybatch file with a case in the database;
- **Status/Priority** – The status category (*no need today, being treated, default, not-active, no answer, busy, busy default, new appointment for today and super appointment*) that indicates how the case can be handled;
- **StartInterval** – The time when a case last became active or, if the case is not active, the time when it will become active;
- **EndInterval** - The time that defines the end point of the range beginning with the time given in the StartInterval;
- **FuturePriority** – The priority category (super appointment, hard appointment, medium appointment, soft appointment or default) that indicates what kind of appointment, if any, the case has pending on the current day;

- **NrOfDials** – The number of separate dial attempts that have been made for the case as part of the current call on the current day. Once a busy dial result is recorded, subsequent consecutive busy dials (up to the maximum number of dials set in the specification file) are not considered to be separate dial attempts;
- **DialInterval** – The time (in terms of the five-minute interval) when the most recent dial attempt on the current day was exited;
- **NrOfDialsBusy** – The number of consecutive busy dial attempts most recently registered for the case;
- **Group** – The name of group to which the case should be routed;
- **Interviewer** – The name of the interviewer to whom the case should be routed;
- **QuotaIndex** – The quota category to which the case belongs;
- **TimeDifference** – The hours and minutes that must be added or subtracted from interviewer time to get current respondent time;
- **SliceID** – The time slice set to which the case belongs;
- **SliceInfo** – The time slice definitions available to the case on the current day.

The primary key is not included with the daybatch file. Instead, each record in the daybatch file is identified by its unique **JoinID**. The **JoinID** is the only one of the 14 fields in the daybatch file that can never be changed.

In order to understand the operation of the Blaise scheduler, you must first understand the difference between the registering of dial results for individual cases and the periodic scheduling of all cases. When an interviewer exits a case, thereby registering a dial result, the values in some daybatch fields are changed for the particular case and only for the particular case. When the entire daybatch is scheduled, Blaise systematically reviews all the cases in the daybatch to identify the cases that should currently be active. As part of this scheduling process, the values in some daybatch fields for a number of different cases may be changed. A table in the Appendix at the end of this document identifies the daybatch fields that can be modified due to scheduling or the registering of a dial result.

When initially generating and then subsequently updating the daybatch file, Blaise must also reference the current settings in the .bts file. This file is known here as “the specification file.” In this technical note, specification file settings are italicized and followed by a bracketed reference to the relevant branch in the specification file. As a general rule, specification file settings can be changed during the interviewing day and the new settings take effect immediately. This applies to all settings in the General Parameter and User branches of the specification file. However, there are a few settings where changes made during the day are not reflected in the daybatch file and hence not applied. In the case of time slices, any change in time slice definitions results in the emptying out of **SliceID** and **SliceInfo** fields for all cases in the daybatch. If this occurs, no time slices are applied until a new daybatch is made.

In this technical note, we will start with the simplest possible base case, where there are no time zones, time slices, routebacks or quotas. After we have reviewed in detail the base case, we will explain what happens in terms of initializing, registering a dial result and scheduling when we add time slices, time zones, routebacks and quotas. Finally we will examine the various priorities in order to explain how Blaise determines what case to deliver to an interviewer.

Note that Blaise divides the day into 288 five-minute intervals, all of which begin when the number of minutes is evenly divisible by five. All times given in the daybatch file are expressed in terms of these five-minute intervals.

For example, if an interviewer exits a case at 9:12 AM, the **DialInterval**, which represents the time exited, is set to 9:10 AM because the call was exited within the 9:10 to 9:15 interval.

Base Case

Initializing Status/Priority, FuturePriority, NrOfDials, NrOfDialsBusy, StartInterval and EndInterval

When the daybatch is first created, all cases are listed with a Status/Priority of *not active*, zero **NrOfDials** and zero **NrOfDialsBusy**. Cases with an exact date and time appointment scheduled for the current day are given a **FuturePriority** of hard appointment. Cases that have a missed exact date and time appointment for the previous calling day and were not contacted are given a **FuturePriority** of medium appointment. All other cases that have relevant appointments for the current day are given a **FuturePriority** of soft appointment, if the current day is not the last possible day for the appointment. They are assigned a **FuturePriority** of medium appointment, if the current day is the last possible day for the appointment. The **FuturePriority** is set at default for all other cases, including cases that have never been called and cases with a no preference appointment.

The default start time is the starting time of the first crew of the day, unless the *Do not call before time [Time zone branch]* is later than this time. In such circumstances, the *Do not call before time [Time zone branch]* is the default start time. Likewise for the end of the day, the default end time is the ending time of the last crew of the day, unless the *Do not call after time [Time zone branch]* is earlier than this time. In such circumstances, the *Do not call after time [Time zone branch]* is the default end time.

When the daybatch is first created, the **StartInterval** and **EndInterval** are determined as a function of **FuturePriority** as follows:

FuturePriority	StartInterval	EndInterval
Default	Default start time	Default end time
Hard appointment	Specified time	Default end time
Medium appointment derived from missed previous day hard appointment	Default start time	Default end time
Medium and soft appointment with specific beginning and ending times	Specified beginning time	Specified ending time
Medium and soft appointments without specific time ranges, excluding medium appointments derived from missed hard appointments	One of the starting crew times, spread out so that such appointments are divided among the various crews in accordance with projected crew size	Default end time

Obtaining a Case

Generally, interviewers obtain cases by running the data entry program (dep.exe). Supervisors obtain cases by running the CATI management program (BtMana.exe). There are four possible ways for an interviewer to obtain a case. These options are available by selecting Forms from the menu bar while running the data entry program. Each option can be shut off. The default option is Get Telephone Number, whereby Blaise uses priority and interviewer-suitability criteria (described below) to determine which is the best case to deliver to the interviewer at the particular time. If the Browse option is available, an interviewer can see all the cases in the database and select whichever case he or she desires. With the Get option, the interviewer can enter the primary key of any of the cases in the database. With the New option, the interviewer can specify a new primary key and enter a new

case in the database. Supervisors obtain cases by selecting a case from the database by using the Forms branch of BtMana.

Once a case already in the daybatch is obtained, its **Status/Priority** is changed to *being treated*. Its **Status/Priority** remains *being treated* until the case is exited. If a case is obtained that is not currently in the daybatch, it is not added to the daybatch until it is exited.

Registering a Dial Result and Exiting a Case

One of eight different dial results must be assigned to a case before the case can be exited. As soon as the case is exited, Blaise re-evaluates the case in the light of the dial result and the nature of the appointment made (if the dial result is appointment). If the dial result is complete, nonresponse, disconnected, other, appointment (not necessarily for today), or answerservice (when *Do not allow multiple same day answering machine calls [General parameters branch]* is checked), Blaise applies its concluding treatment. For cases that are already in the daybatch, the concluding treatment means that their **Status/Priority** becomes *no need today* and their **NrOfDials** is incremented by one.

If the dial result is noanswer or answerservice (when *Do not allow multiple same day answering machine calls* is not checked), Blaise assigns the **Status/Priority** of *no answer*, but makes no other changes upon exiting the case. If the dial result is busy, Blaise assigns the **Status/Priority** of *busy*, but likewise makes no other changes upon exiting the case. If the dial result is appointment and the appointment is a hard appointment for later in the day or a medium appointment for which the current day is the only possible day, Blaise assigns the **Status/Priority** of *new appointment for today* and changes the **FuturePriority** to either hard appointment or medium appointment.

As part of the process of registering a dial result, the five-minute time interval in which each case is exited is stored in its **DialInterval** field. The time on the interviewer's workstation is used, so it is important to synchronize times on all workstations and servers.

The table below summarizes the adjustments, which depend upon the dial result and are made in the daybatch file for each individual case when it is exited. A series of dashes indicates that no changes are made when the case is exited. Some of these fields are changed, however, when the case is subsequently scheduled.

Daybatch field	Concluding dial result	Appt not necessarily for today	Appt necessarily for today	Noanswer or answerservice	Busy
Status/Priority	No need today	No need today	New appt for today	No answer	Busy
StartInterval	-----	-----	Starting time for appt	-----	-----
EndInterval	-----	-----	Ending time for appt (if a range of times is given) or default end time	-----	-----
FuturePriority	-----	-----	Hard appt or medium appt	-----	-----
NrOfDials	Incremented by 1	Incremented by 1	Incremented by 1	-----	-----

No entry is made in the daybatch file for cases that receive Blaise's concluding treatment and are not already in the daybatch. However, a new entry is made at the end of the daybatch file for all cases that are not already in the daybatch to which the concluding treatment is not applied. The only exception to this rule occurs when *Apply select fields also after the dial in the DEP [Daybatch select branch]* is checked (enabled). In these circumstances,

a case cannot be added to the daybatch if the values in its select fields (those fields to which selection criteria are applied during daybatch creation) are such that it could not be included in the daybatch.

Registering a Super Appointment

At any point after the daybatch is created, a supervisor using BtMana can give a super appointment to a case. This can be done by:

- Viewing a list of all cases in the Forms branch;
- Double clicking on the particular case; and
- Selecting Call as soon as possible, from the dial menu.

Giving a case a super appointment is considered a different activity from obtaining a case. No dial result is assigned and the number of calls for the case is not incremented. If a super appointment is given to a case that is not in the daybatch, the case is automatically added to the end of the daybatch.

When a super appointment is registered for a case, its **Status/Priority** code is set to *super appointment* and its **FuturePriority** is set to super appointment. Its **DialInterval** and **StartInterval** are set to the five-minute interval in which the super appointment was made. Its **EndInterval** is set to the default end time. Both its **NrOfDials** and **NrOfDialsBusy** in recent versions of Blaise are set to zero.

Activating Cases

When there are no cases currently available for delivery, or at any other time after a daybatch is created, a supervisor can use BtMana to activate all the cases in the daybatch. Supervisors can do this by:

- Highlighting the View Daybatch branch;
- Selecting Management from the menu bar; and
- Selecting Activate.

A supervisor can also activate an individual case already in the daybatch by:

- Highlighting the Forms branch or by highlighting the Browse Daybatch sub-branch in BtMana;
- Double-clicking on the specific case;
- When the Case Summary dialog appears, clicking on the More button; and
- When the More info dialog appears, clicking on the Activate button.

Only cases with a **Status/Priority** of *not-active*, a **FuturePriority** of default, soft or medium appointment and 0 **NrOfDialsBusy** can actually be activated. When a case is activated its **StartInterval** is changed to the current five-minute interval and its **Status/Priority** is changed to *default*. None of its other daybatch fields are changed.

Scheduling the Daybatch

Blaise schedules the daybatch the first time that an interviewer requests a case or exits a case at the beginning of a five-minute interval. It does this by re-evaluating all the cases that do not have a **Status/Priority** of *being treated*, *busy default*, *super appointment* or *no need today*. As part of this re-evaluation process, Blaise may possibly change the **Status/Priority**, **FuturePriority**, **NrOfDials** and **NrOfDialsBusy** for the case. Scheduling can occur at any point if a supervisor selects the Schedule option using Btmana. The scheduling process can be divided into three steps.

Step 1: Adjusting the number of dials

Changes are made in the number of dials for cases with a **Status/Priority** of *new appointment for today*, *no answer* and *busy* as shown in the below table.

Daybatch Field	New appointment for today	No answer	Busy
NrOfDials	Set to 0	Incremented by 1	Incremented by 1 if and only if NrOfDialsBusy = 0 before scheduling begins
NrOfDialsBusy	Set to 0	Set to 0	Incremented by 1 and then reset to 0 if maximum number of busies is reached

The maximum number of busies (these busies must all be consecutive busies) is reached when the **NrOfDialsBusy** reaches the *Maximum number of busy dials [General parameters branch]*. At this point, Blaise essentially decides to stop chasing busies and aggregates all the consecutive busies into the equivalent of one dial. Because the **NrOfDials** is incremented at the same time as the first busy is registered, Blaise does not increment the **NrOfDials** at the time that the maximum number of busies is reached (unless of course the maximum number of busies is set to one).

If after these adjustments the **NrOfDials** reaches the *maximum number of dials [General parameters branch]*, the **Status/Priority** goes to *no need today*. If this happens, steps 2 and 3 below are skipped.

Step 2: Determining a new StartInterval

A new **StartInterval** needs to be determined for all cases with a **Status/Priority** of *no answer* or *busy*. In order to do this the minimal possible spread needs to be obtained from the specification file. For cases with a **Status/Priority** of *no answer* and a **FuturePriority** of hard or super appointment, the minimal possible spread is equal to the value given in *Minimum time between hard/super no answer [General parameters branch]*. For all other no answer cases, the minimal possible spread is equal to the value given in *Minimum time between other no answer [General parameters branch]*.

For cases with a **Status/Priority** of *busy*, if the **NrOfDialsBusy** equals one, the minimal possible spread is equal to the first of the eight different values stored in the *Minutes between busy dials [General parameters branch]* settings. If **NrOfDialsBusy** is 2, then the minimal possible spread is equal to the second value stored in *Minutes between busy dials [General parameters branch]* settings and so on. If the **NrOfDialsBusy** equals zero, the minimal possible spread is determined in the same way as it is for a **Status/Priority** of *no answer*.

The minimal possible spread is added to the **DialInterval** to get the earliest possible start time. If the earliest possible start time is equal to or later than **EndInterval**, the **Status/Priority** goes to *no need today*. If the earliest possible start time is before the **EndInterval**, the **StartInterval** is set equal to the earliest possible start time for cases with a **FuturePriority** of default, hard appointment or super appointment. The determination of the **StartInterval** is somewhat more complicated for cases with a **FuturePriority** of medium or soft appointment, because in addition to honoring the minimal possible spread, the scheduler tries to spread call attempts out more or less evenly throughout the possible calling period.

One exception is made to the rule that if the earliest possible start time is equal to or later than **EndInterval**, the **Status/Priority** goes to *no need today*. This exception occurs in the case of medium appointments with specified time ranges on the last day of the survey. Blaise allows the equivalent of a new call to be applied to these cases. It resets both the **NrOfDials** and **NrOfDialsBusy** to zero, while setting the **FuturePriority** to default, the

Status/Priority to *default*, the **StartInterval** to the current five-minute interval and the **EndInterval** to the default end time.

Step 3: Comparing current time with **StartInterval** and **EndInterval**

As a final step, the **StartInterval** needs to be compared with the current time for cases whose **Status/Priority** has not already gone to *no need today*. If the current time is before the **StartInterval**, the **Status/Priority** is set to *not-active*. If the current time is between the **StartInterval** and the **EndInterval**, the **Status/Priority** is set to *default* or *busy default*, if the **NrOfDialsBusy** is greater than zero. If the current time is equal to or later than the **EndInterval** (this can happen only in the base case when there are soft or medium appointments with specified time ranges), the **Status/Priority** goes to *no need today*.

Adding Time Slices

Initializing SliceInfo, SliceID, StartInterval and EndInterval

When the daybatch is first created, the **SliceID** field is automatically filled in with the appropriate time slice code. In some surveys a field in the database is given time slice functionality in the specification file. In these instances the string stored in the database in the field with time slice functionality, is written into the **SliceID** field. If the *Time Slice* field is blank or unrecognizable in the database, the *Slice ID* field is left empty in the daybatch file.

If no field has been given time slice functionality, the code of the first time slice set, defined in the specification file, is written into the **SliceID** field, for all cases. If no time slice set has been defined in the specification file, the **SliceID** field is empty.

If the **SliceID** field is filled in, the **SliceInfo** field must be filled in with the numbers of all available time slice definitions for the current day. The time slice definitions for any given time slice set are numbered in order, starting with 1, for the first definition given in the specification file. A time slice definition is available if it is relevant for the current day and has yet to be tried the number of times specified for it. For example, if the first definition is for Mondays from 2 PM to 5 PM with 2 tries and today is Monday, the SliceID of 1 can be listed only if the case has never been called or only called once on a Monday between 2 and 5 PM.

If time slices are applicable, the **StartInterval** is set to the beginning time of the earliest time slice definition in the **SliceID** field and the **EndInterval** is set to the ending time of this time slice definition. Time slices are basically designed to spread out no answer (and answerservice) attempts. They are only applicable for cases with default **FuturePriority** whose most recent dial result is noanswer or answerservice. They are not applicable for cases that have never been attempted. If the **StartInterval** and **EndInterval** are set in accordance with a time slice definition, a minus sign in parenthesis is added at the end of the SliceID field.

Time Slice Applicability During Scheduling

Time slices become inapplicable when any case with a **Status/Priority** of *new appointment for today*, or *busy* is scheduled. When this happens, the minus sign, if it exists, is removed from the **SliceID** field. Alternatively, time slices become applicable for any case with a **Status/Priority** of *no answer* with a default **FuturePriority**. When this happens during scheduling, a minus sign, if it does not already exist, is added to the end of the **SliceID** field.

Adjusting **StartInterval** and **EndInterval** During Scheduling

When time slices are applicable, the **StartInterval** and **EndInterval** must be changed whenever a case with a **Status/Priority** of *no answer* is scheduled. If the relevant time slice set only allows one try per day, the case automatically goes to *no need today*. Even if more than one try is allowed per day, only one try per time slice definition is permitted per day. This means that once a time slice definition is tried, Blaise must find a new time

slice definition in which the case can be delivered. In order to do this, Blaise adds the *Minimum time between other no answer [General parameters branch]* to the **DialInterval** to determine the earliest possible start time. Blaise then determines which of the available time slice definitions in the **SliceInfo** field start at or after the earliest possible start time. It takes the earliest of these time slice definitions and sets the **StartInterval** and **EndInterval** accordingly. If there are no available time slice definitions that start at or after the earliest possible start time, the **Status/Priority** goes to *no need today*.

When time slices are applicable, the **StartInterval** and **EndInterval** may also have to be changed for cases with a **Status/Priority** of *default* or *not-active*. This happens if the current time is equal to or later than the **EndInterval**. In these circumstances, Blaise determines if there is an available time slice definition that includes the current time and sets the **StartInterval** and **EndInterval** accordingly. The **Status/Priority** is correspondingly set to *default*. If there is no available time slice definition that includes the current time, Blaise finds the earliest possible next available time slice definition and sets the **StartInterval** and **EndInterval** accordingly. The **Status/Priority** is set to *not-active*. If no appropriate time slice definition is available, the **Status/Priority** is set to *no need today*.

Changing Value of Time Slice Field in the Database

If the value in the time slice field is changed in the database, the **SliceID** and **SliceInfo** fields are not changed in the daybatch file. Blaise continues to reference the information in the daybatch file. Changes to the time slice field in the database take effect the next time the daybatch is created.

Adding Time Zones

Initializing TimeDifference

If a field is given time zone functionality in the specification file, the **TimeDifference** field is filled in when the daybatch is first created. Blaise enters the difference (in hours and minutes) given in the specification file for whatever time zone appears in the time zone field in the database. If the time zone field in the database is empty, or the same as interviewer time, the **TimeDifference** field is left empty.

Adjusting Times

All times used in the daybatch file are in interviewer time; however appointment times [derived from running the data entry program], time slice definitions [*Time slice branch*] and the no call limits [*Time zone branch*] are in respondent time. This means that the hours and minutes given in the **TimeDifference** field must be subtracted from the respondent times before these times can be used in the determination of the interviewer-based **StartInterval** and **EndInterval**. When a minus sign is present in the **TimeDifference** field, the subtraction of a negative value basically means that the **TimeDifference** is added to the respondent time to get the interviewer time.

For example, let us take a situation with three contiguous time zones. Time zone A is 60 minutes ahead of time zone B, which in turn is 60 minutes ahead of time zone C. Interviewer time is time zone B time. A time slice definition defined in the specification file, or an appointment range defined in database as 4 PM to 6 PM, would result in the daybatch file of a **StartInterval-EndInterval** range of 3 PM-5 PM for time zone A, 4 PM - 6 PM for time B and 5 PM to 7 PM for time zone C.

Changing Value of Time Zone Field in the Database

If the value in the time zone field is changed in the database, the **TimeDifference** field is not changed in the daybatch file. However, Blaise routinely references the database to ascertain how much of a time difference adjustment, if any, should be made to appointment times, time slice ranges and no call time limits.

Adding Routebacks

Initializing Interviewer and Group

If a field is given routeback functionality in the specification file, when the daybatch is first created, the value in the routeback field is written into the **Interviewer** field, if this value is listed as the name of an interviewer in the Users/Interviewers branch of the specification file.

Alternatively, the value in the routeback field is written into the **Group** field, if the value in the routeback field in the database is listed as a group in the Users/Groups branch of the specification file. If the **Interviewer** field is filled in and groups have been defined in the specification file, the name of the interviewer's main group is written into the **Group** field.

Changing the Interviewer and Group

The value stored in the routeback field in the database for any given case, can be changed while running the data entry program. It can also be changed when using BtMana to treat a case, by assigning a different interviewer or group to the case. Finally the value stored in the routeback field can be changed if *Route back to interviewer [General parameters branch]* or *Route back to group [General parameters branch]* are enabled in the specification file. If *Route back to interviewer [General parameters branch]* is enabled whenever an appointment is made for a case, the name of the interviewer making the appointment is automatically filled into the routeback field. Likewise if *Route back to group [General parameters branch]* is enabled the name of the main group of the interviewer making the appointment is automatically filled into the *routeback* field. If the value stored in the *routeback* field in the database is changed, and the case does not go to a **Status/Priority** of *no need today*, the change in value is also reflected in the **Interviewer** and/or **Group** fields in the daybatch.

Restricting Delivery in Accordance with the Values in the Interviewer and Group Fields

Cases with a **FuturePriority** of default or soft appointment are only delivered to the interviewer named in the **Interviewer** field. If the **Interviewer** field is empty, but the **Group** field is not empty, default and soft appointment cases are only delivered to interviewers that are listed in the specification file as belonging to the specified group. If both the **Interviewer** and **Group** fields are empty, cases with a **FuturePriority** of default and soft appointment can be delivered to any interviewer.

Blaise relaxes the restrictions that apply to default and soft appointment cases for cases with medium, hard and super appointments. It does this by letting users set expiration delays in the specification file. In the case of medium, hard and super appointments, both the *Interviewer de-activation delay [General parameters branch]* and the *Group de-activation delay [General parameters branch]* can be set at anywhere from 5 minutes to 1440 minutes (24 hours). If *Expire on de-activation delays only [General parameters branch]* is checked in the specification file, an *Interviewer de-activation delay [General parameters branch]* of 10 minutes means that the routeback to the specified interviewer expires 10 minutes after the starting time of the appointment. Once this delay has expired, an asterisk appears after the name of the interviewer in the daybatch file. As long as the delay has not expired, the case can be delivered only to the specified interviewer.

If the *Group de-activation delay [General parameters branch]* is set at 20 and no interviewer is specified in the **Interviewer** field, the routeback to the group expires 20 minutes after the starting time of the appointment. However, if an interviewer is specified in the **Interviewer** field, the interviewer delay must expire first, before the 20-minute group delay is applied. Once the group delay has expired, an asterisk appears after the name of the group in the daybatch file. Note that the specification file allows for two different sets of interviewer and group

expiration delay. One set applies to medium appointments, while the other set applies to both hard and super appointments.

The *Expire on de-activation delays only [General parameters branch]* setting in the specification file plays a critical role in determining if a routeback delay has expired. If this setting is not checked and therefore not enabled, expiration delays are only relevant if the specified interviewer, or someone from the specified group are actually involved in interviewing. For example, if the group de-activation delay is 20 minutes, but no one from the group is involved in interviewing, Blaise expires the group de-activation delay.

Adding In Quotas

Initializing QuotaIndex

If one or more fields are given quota functionality in the specification file, when the daybatch is first created, the quota category to which each case belongs is written into the **QuotaIndex** field. Integers are used to represent quota categories. If you select the quota control tab in the specification file, you can see all the quota categories. The integer 1 is given to the first category listed, the integer 2 to the second category and so forth. If not enough information exists in the database to assign a case to any quota category, a zero is written into its **QuotaIndex** field.

Filling a Quota

Cases that belong to a quota category that has already been filled are not included in the daybatch. If during the course of an interviewing day, a case is completed that brings the number of completed cases for its quota category up to the count targeted in the specification file, all other cases in the daybatch from this quota category with default **FuturePriority** are automatically given the **Priority/Status** of *no need today*. This quota-based adjustment occurs as soon as the triggering case is exited and not as part of the scheduling process.

Modifying the QuotaIndex

If enough information is gathered on a case during the interviewing day to assign it to a quota category or to change its quota category, the new quota category is written into the **QuotaIndex** field as soon as the case is exited. If any case is attempted but cannot be assigned a quota category, -1 is written into its **QuotaIndex** field.

Priorities (Determining Which Case to Deliver)

When an interviewer requests a case, Blaise systematically goes through all the cases in the daybatch that can potentially be delivered to the particular interviewer. Any case whose **Status/Priority** is currently *default*, *super appointment* or *busy default* is potentially deliverable. In order to determine if a case can be delivered to a particular interviewer, Blaise looks in the daybatch file at the names in the **Interviewer** and **Group** fields, as well as at the **FuturePriority**. A case can be delivered to a particular interviewer if:

- The interviewer's name is in the **Interviewer** field;
- The **Interviewer** and **Group** fields are both empty;
- The **Interviewer** field is empty and the interviewer is a member of the group listed in the **Group** field;
- The **FuturePriority** is medium, hard or super appointment and an asterisk (for an expired delay) appears in both the **Interviewer** and **Group** field or an asterisk appears in one of these two fields and the other field is empty; and

- The **FuturePriority** is medium, hard or super appointment and an asterisk (for an expired delay) appears in the **Interviewer** field and the interviewer is a member of the group listed in the **Group** field.

Blaise assigns one of eight different priorities to each case that can be potentially delivered to the particular interviewer. These priorities in order are:

1. Super
2. Hard-busy
3. Hard
4. Medium-busy
5. Soft-busy
6. Default-busy
7. Medium
8. Soft
9. Default

Super, hard, medium, soft and default priorities are derived from a **FuturePriority** of super appointment, hard appointment, medium appointment, soft appointment and default, respectively. The term busy is only added to the priority if the last dial result on the current day was a busy and not the last possible busy in a consecutive set of busies (i.e. the **NrOfDialsBusy** is greater than zero)

Once Blaise determines the highest priority possible for an interviewer, it focuses only on the cases at this priority level. For example, if there are no cases with super priority but one with hard-busy priority, it delivers this particular case to the interviewer.

If there is more than one case potentially deliverable to a particular interviewer within the highest available priority, Blaise chooses the case to deliver by ascertaining the most suitable case for the interviewer.

Suitability, from most suitable to least suitable, is determined as follows:

- Interviewer's name is in the **Interviewer** field;
- Interviewer's main group is in the **Group** field;
- One of the interviewer's secondary groups is in the **Group** field;
- The case can go to anyone.

Suitability is determined irrespective of whether delays have or have not expired. This means that even if a delay has expired, interviewers get cases meant for them before they get cases at the same priority meant for other interviewers.

If several cases are equally suitable at the highest available priority level, Blaise delivers the case with the lowest **NrOfDials**. If several such cases share the lowest **NrOfDials**, Blaise delivers the case with the earliest **StartInterval**. If several such cases share the same **StartInterval**, Blaise delivers the case that is listed first in daybatch file.

Appendix

Daybatch Field Re-evaluation by Activity

	Registering a Dial Result	Scheduling	Activation of All Cases	Registering a Super Appointment
JoinID	Never	Never	Never	Never
Status/Priority	Always	Sometimes	Sometimes	Always
StartInterval	Sometimes	Sometimes	Sometimes	Always
EndInterval	Sometimes	Sometimes	Never	Always
FuturePriority	Sometimes	Never	Never	Always
NrOfDials	Sometimes	Sometimes	Never	Always
DialInterval	Always	Never	Never	Always
NrOfDialsBusy	Never	Sometimes	Never	Always
Group	Sometimes	Never	Never	Never
Interviewer	Sometimes	Never	Never	Never
QuotaIndex	Sometimes	Never	Never	Never
TimeDifference	Never	Never	Never	Never
SliceID	Never	Never	Never	Never
SliceInfo	Never	Never	Never	Never