Newsletter #28 (July 19, 2007)

Contents

What's New

Newsletters are back!

We apologize for the long delay since the last newsletter, and thank you for all of the "we miss them" messages. The fall & winter season is our busiest time, and we have been busier than ever (thank you for helping to spread the word about Campground Master!). This along with trying to get a new version completed with much-requested features and Vista support has kept us from getting a newsletter out.

We hope you find this extra-large newsletter useful, and we'll do our best to keep them coming!

New version 4.2.2

A new version 4.2 was released last month, along with follow-up patches 4.2.1 and 4.2.2 to fix some issues discovered in the new release. These are relatively minor updates with various bug-fixes and some much-requested enhancements.

If you're a current customer then you should have already received the notices by E-mail -- if you didn't, then please contact us to update your E-mail address information. If you're using a demo version then you can still download the new version, just be sure to use the Backup function to save your database if it's still called

"Demo" (the demo database is replaced when upgrading the software).

Here are the most noteworthy changes:
- Changed the Help file format and database locations to be compatible with Windows Vista.
- The "Purge Old Data" function has been improved so it doesn't have memory usage problems.
- Credit cards numbers are masked so low-level operators can't see them.
- Non-Synchronized can now be the default for linked reservations.
- Linked group members can now be checked in simultaneously from separate workstations.
- SMTP server settings can be separate for each workstation
- Multiple credit card slips can be printed automatically
- A reservation receipt can be printed automatically when a payment is made
- U.S. Zip codes & Canadian postal codes data has been updated
- Fixed cash drawer opening issues when using a receipt printer driver
- Fixed landscape/portrait report printing issues (printing the wrong # of pages)

For upgrading information, see our web site:
http://www.campground-master.com/download_upgrade.html

Windows Vista Compatibility

If you're getting a new computer, it will probably have Windows Vista loaded on it. This new operating system still has a lot of quirks in it that many people don't like (so we suggest staying with XP if possible), but it's difficult to avoid it in new computers.

The main question of course is whether Campground Master works with Vista. There were a few issues with versions earlier than 4.2, so workarounds are posted on the "System Requirements" page of our web site. However the new 4.2 version is fully Vista compatible.

One issue that may remain is the compatibility of external hardware -- specifically, 3" receipt printers with a USB interface. The Epson receipt printers, for instance, don't currently have drivers available for Windows Vista. However if you're using a parallel or serial interface and the "direct to port" option then no driver is needed, and those will still work with Vista.

Expression text parsing change in version 4.2

In order to correct a problem with "escape sequence" parsing in text expressions, a change had to be made which might impact existing expressions.

NOTE: This only affects the Advanced Customizations such as Queries, Events, Dialogs, etc. that you might have added yourself, so this change won't affect most users.

The change is that escape-sequences like \r and \n in quoted text are now converted in the parsing stage rather than the execution stage. Thus any text variables, data fields, etc. are considered already converted when passed to the functions MessageBox and SendMail (these are the only two functions that this directly affected, at least for the \r and \n character escapes). The primary effect of this on existing code is that if a backslash character is used in nested quotes (either for escapes or for other things like file paths), it must be double-escaped now.

Here is an example -- in previous versions, this expression would work:

```
EvalQ( 'MessageBox("First line \n Second line")')
```

Now in version 4.2, the parameter 'MessageBox(...)' inside EvalQ is parsed as quoted text before execution by EvalQ, so the \n is converted before parsing the parameter of the MessageBox function and the resulting line feed would break the expression (a line feed signals the end of an expression).

In version 4.2 and later, you need to change it to have two backslashes so that the "\\" is converted to "\" :

    EvalQ('MessageBox("First line \\n Second line")')

Thus after parsing first set of quotes it will result in the correct expression:

    MessageBox("First line \n Second line")

And then of course the text will be given to MessageBox with a proper line feed in it (\n is converted to a line feed), resulting in 2 lines of text in the message box.

In summary, if you have any text in expressions that contain backslash characters, and if the expression contains more than one level of quotes, you need to double up the backslashes for each level of quotes.

Optimization for busy POS environments

If you use the POS functionality and have some items that are sold very frequently like ice or day passes, there can be a significant a bottleneck in the program after awhile -- especially noticeable in the form of network delays.

This is a result of a field in the inventory items data which keeps a list of all transactions done for each item. This list can of course get very large, with thousands of links in it, and this entire list has to be updated and synchronized each time the item is sold. The temporary solution has been to use Purge Old Data to minimize the size of the field, which can help quite a bit but is not ideal.

In version 4.2.2, we made it possible to eliminate this field so there is no impact on speed when doing sales. All you have to do is go into Maintenance / Data Field Definitions / Inventory Item Fields, and change the status of the "Inv_Link_List_Trans" field to "Disabled".

HOWEVER, you should be aware of these side effects:

1. This is not reversible -- although the field can be re-enabled, all previous data in it will be lost (causing the reports mentioned below to be inconsistent).

2. This will cause the Old Stock report to be extremely slow -- it may take several minutes to bring it up (and if you try to sort by Date Last Sold, it may take hours).

3. Other POS reports that allow filtering inventory items will also be very slow if the Inventory Filtering includes a filter by date sold or number of days since last sold.

If these report issues are not a concern, then we recommend disabling that field as mentioned to minimize network delays. In future versions we will use dynamic indexing of the transactions so that the reports won't be slow, and at that time the field will be completely removed anyway.

Q & A

Viewing the full credit card number

Q: After upgrading I can only see the last 4 digits of credit card numbers. How can I see the whole credit card number again?

A: One of the changes in version 4.2 is that credit card numbers are "masked" for low-level operators -- only the first digit and last 4 digits of the credit card information are visible after it's initially entered. This was changed to increase security and for compliance with merchant rules.

Actually the masking is simply on by default for operators which don't have Administrator access, so the simple workaround is to log in as an administrator when needed. Administrators will still see the full number (so of course it's important to make sure that clerks don't know an Administrator level login).

One other exception is that the credit card number in the Reservation Guarantee Info will be visible by Manager level operators as long as the reservation is not yet checked in or cancelled, so for instance a cancellation charge can be put on the card by a manager if needed. Be sure to get the card information before you actually cancel the reservation, though, since it's masked after cancellation.

You can also change these access levels if you prefer -- just be aware of the security implications if you do. In Maintenance / Park Setup / Access Levels, look for the series of functions that start with "CC number masking --" and change them as needed.

Note that if you're processing credit cards through Campground Master then it's not necessary to see the full card number in order to put a Payment or Deposit on that previously entered card, in which case there's little reason for anyone to be able to see the full card number.

Printing only some transactions

Q: I want to print just the last few transactions for a reservation. When I click "Print only some transactions", it just prints the last one which isn't enough. How do I print more without printing all of them?

A: Note the "Print:" options next to that button. There are 4 options for designating what you want to print (New, Since last zero-balance, etc.). Most likely you can use the "Since date" option and enter an appropriate date. The other options might be useful for monthly renters, etc. More detailed information on these options can be found in the documentation.

Re-using a CD-RW for backup

Q: The first time I backed up to a CD-RW disk it worked fine. Now when I try it I get an error message. Isn't it supposed to be able to re-use the CD?

A: You need to either enter a different file name each time you make a backup to the same CD, or else re-format the CD (through Windows). Unlike floppies or USB flash memory drives, it's generally not possible to overwrite an existing file on a CD (with the same name) without reformatting it first.

If using a different name doesn't help, it might also be that Windows is configured to "Close" the disk each time, preventing further re-writes. Again, re-formatting the CD should work.

Removing unused receipt formats

Q: We only use a couple of the receipt formats shown in the drop-down list (in Reservation Transactions for instance). How can I remove the ones I don't need?

A: If you have version 4.0 or later, these are called Forms -- you can disable the unused Forms so they don't show up (you can also delete them, but we don't recommend this in case you want to use them later).

Go to Maintenance / Advanced Customizations / Forms. For each of the receipt formats you want to hide, select it and "Edit form definition" (or just double-click on it), then uncheck the "Enabled" box in the Edit Form Definition dialog, and click Save. Note that mailing labels, envelopes, and E-mail confirmations are also listed here but they won't show on the drop-down receipts list, so don't disable those by mistake.

Marking sites for special use

Q: We want to flag some sites for use only under certain conditions, like a special event. We need them to be available for reservations, but we want to show them with an event name or special color so the operators know they aren't to be used for normal reservations. How can we do that?

A: Probably the best way to do this is to put an Owner reservation on the sites, with the name of the event. Owner reservations will show up on the Rack (assuming you have that option checked in the Options on the Rack), but will still allow normal reservations to be made on top of them. It's as if the event "owns" the sites for that period, but they can still be rented out.

First you need to make sure that at least one Owner reservation type is enabled -- see Maintenance / Pick Lists / Reservation Types. If you prefer, you can add a special reservation type for the event or rename the Owner type -- just make sure the Base type is "Owner".

Now make a reservation on the site like you normally would, but select Owner (or your special type) form the "Resv Type" list. Make sure the Last name of the customer (e.g. a fake customer) is what you want to show on the Rack.

To make the real customer reservations in those sites, you can double-click in the space like you normally would to start a reservation, or right-click and select "New Reservation for this site (non-owner)". Selecting the site through New Reservation works also.

Printing labels in Last Name order

Q. How can I print mailing labels in order by name instead of Zip code?

A: You simply need to sort the list before printing the labels. Many "lists" in Campground Master are sortable by any column. In the Print Labels dialog, click on the Last Name column heading (e.g. click on "Last Name"). This will sort it, or if it's already sorted, will change the direction of sorting.

Now when you print the labels, they will be in the same order as shown in the list.

Getting a history report for a site

Q. Is it possible to get a history report for a specific campsite?

A. On the Rack, right-click on the site (in the left-hand column) and select "Find all reservations for this site". This will show all reservations ever made for the site.

How to keep from entering Customer Transactions

Q: Sometimes a clerk enters a payment or deposit through Customer Details instead of through the Reservation. I know we aren't supposed to do that, but is there any way to keep them from doing it by mistake?

A: You can set the Access Level so that only Administrators or Managers are allowed to enter Customer transactions.

Go to Maintenance / Park Setup / Access Levels. Look for the entry "Enter Customer (non-reservation) transactions", and double-click on it. Then change the New Minimum Access Level as desired.

How to find a reservation by phone number

Q: How can I find a reservation by phone number?

A: The phone number is part of the Customer information, not the Reservation information (refer to the documentation for a description of relational database organization) -- so you can't directly use Find Reservation for this. Instead, go to Customers / Find Customer. This function allows you to search by phone number as well as several other fields (selected in the "Filter by" drop-down list). Once you find the customer, then you can get to the reservation (in Customer Details, click Reservation History).

Tip: when searching by phone numbers, only enter the digits without punctuation. When searching it ignores non-digits, so for instance "3015551212" will find a phone number even if it's entered on the customer details as "Home: (301) 555-1212". However this special case of a digits-only search is only for phone numbers.

Changing the access level for printing tab views

Q: I want clerks to be able to print the tab views, but I can't find an Access Level setting for this. Can I change the levels for printing?

A: (Assuming version 4.0 or later...) Go to Maintenance / Advanced Customizations / Tab Views. Edit each tab view and change the "Access level required to print" as needed.

Tips & Techniques

Making use of sites when the annual leaser is away

When an annual leaser or seasonal camper is not present on his site, you may want to maximize income by using that site for transient customers. Of course Campground Master would normally not allow double booking, so you need to be able to temporarily remove the long-term reservation from the site or otherwise get around this restriction.

There are a few ways to handle this....

You could use "Guest" reservations for the transient renters, since Guests can be on top of a normal reservation. (On the Rack, you can right-click and use "Add a guest reservation".) This is more of a quick-and-dirty method, which is OK if it's a rare occurrence. You can't see guest reservations on the Rack, but they do appear on the other reports and tab views. You might also need to adjust your Rate definitions to handle Guest reservation types, and of course being a different reservation type could affect other functions and filtering.

You could also "split" the seasonal reservation and adjust the first night of the new part for when they're returning, essentially creating a gap in the original reservation that can be used for transients. This is a fairly clean method (it results in 2 linked reservations), but it can cause problems with rate auto-calculations since the reservation is no longer continuous.

Usually a best way is to use the "Owner" reservation type for the annual reservations. This type of reservation is designed for site owners or annual leasers that may not be present the whole time -- the Owner reservation doesn't actually "occupy" the site, e.g. it's still considered "available" for normal reservations, but allows tracking of the rent and other fees for them.

When using an Owner reservation, you should also make a "Normal" (or perhaps "Free Stay") reservation for the same person for the time that they are actually there. While this does mean more work to track when the leaser is on-site (which is actually useful sometimes), it also allows the site to be used normally for transients when the site isn't actually occupied by the leaser.

For more details about owners, see the previous newsletter article here:
http://campgroundmaster.com/news/ownerscantbecheckedin.html

Credit card "swiped" vs "non-swiped" fees

Most credit card merchant accounts have lower transaction fees (or "discount" percentage) when a credit card is present as opposed to not-present (payment by phone or online). The way they know whether it's present or not is by extra "swipe" data that's read from the card when you physically swipe the card through the credit card terminal or computer-based swipe device.

This difference in fees applies whether you process cards through Campground Master or though a separate terminal. Here are a couple tips for minimizing the fees when processing them through Campground Master, but they can also be useful or separate terminals.

First of all, if you don't need to actually charge the card when they call to make the reservation but you want a card on file in case they cancel, then be sure to use the "Guarantee" function so you can enter the card information in the appropriate fields. This is better than just using the Notes part of the reservation or guarantee info, for security and for ease of charging the card later.

Now the important part to remember is that whether or not you charged a deposit by phone, even if they want you to put it on the same card, don't just use the card information you manually entered because it doesn't have the swipe data. When they arrive, ask to swipe their card when paying for any balance or new charges (click "Re-do swipe" on the Enter Payment screen to clear out the non-swiped information). This way it will count as a "card present" sale and you will avoid the higher "card not present" fees.

Once the card is swiped, Campground Master will retain that swipe data for any further credit card payments on the same reservation in case they incur additional charges during their stay -- but for future reservations it will pull the non-swiped information from the "Guarantee Info" on the Customer Details and/or Reservation Details. So you should still swipe their card each time they arrive.

Automatic logout after 10 minutes

We're often asked about adding a feature that will automatically log out the last operator if the computer is inactive for a period of time. In version 4.0 and later, this can be done by adding a simple Event Action. This example will log out after 10 minutes, but you can change the time by altering the number of seconds in the function.

1. To add this feature, go to Maintenance / Advanced Customizations / Event Actions.

2. Click Add Event Action.

3. Enter an Action Name, like "Logout after 10 minutes".

4. For the Event Trigger, select "Each Minute".

5. For the Condition, enter the following (all in one line -- if this shows it as multiple lines, enter it continuously without pressing Enter).

```
CurrentOpLevel() > 0 AND !IsAnyDialogOpen() AND (SecondsInDate() -
SecondsInDate(TimeOfLastInput())) > 600
```

What this does is make sure someone is already logged in, no dialogs are currently open, and it has been 600 seconds or more since the last time any mouse movement or key pressed.

6. For the Action expression, enter:

```
LogOut()
```

7. Click Save, then Close to exit Event Actions Setup.

Note that since this event is only checked each minute, the actual inactivity delay may be anywhere from 10 to 11 minutes. You could make it trigger each second, but that would result in more overhead than necessary for something that doesn't need to be so precise.

Adding check-date fields to dialogs

When adding fields to Dialogs through Advanced Customizations, much of the expression work is done for you using the "Quick-Add Fields" function. However you'll notice that when adding date fields, this shortcut function only adds it as a simple date control and not a "check-date" control (as seen for fields like Blocked-To" and "Paid Thru". If you want a date field that can be "blank" instead of always having a date in it, you need to use a date control with a checkbox.

This example shows how to convert a simple date control to a check-date control.

The example below assumes you have a field called "Cust_User_Date" that you're adding to the dialog, and the field name is "Date" so that the resulting date selection control is called "Date Input" (you'll need to change these in your expressions as needed for your names, of course).

After using Quick-Add to add your date field, edit the "Date selection" element that's automatically added and follow these steps to make it a check-date field:

1. Change the Control Type to "Date selection with a checkbox".

2. Change the Initialize expression so that it sets the checkbox according to whether the field is blank or not. For instance:

```
IIFQ(FieldIsBlank(ThisCust(),"Cust_User_Date"), 'DlgSetUserCtrlDateCheck("Date Input",
.F.)', 'DlgSetUserCtrlDate("Date Input", FieldDate(ThisCust(),"Cust_User_Date"))')
```

3. Change the OK/Save expression to set the field to blank if the box is not checked:

```
IIFQ(DlgGetUserCtrlDateCheck("Date Input")=.F., 'SetFieldBlank(ThisCust(),
"Cust_User_Date")', 'SetFieldDate(ThisCust(), "Cust_User_Date", DlgGetUserCtrlDate("Date
Input"))')
```

That's all there is to it, so Save the changes to the element. Now it will act like other check-date controls and allow the field to be blank (no date set) when the checkbox is not checked.

Adding custom pick-lists to dialogs

When adding custom data fields, you may want to add a data field and dialog selection list that only has certain choices that the operator can choose from, the way Campground Master uses Pick Lists.

While it isn't possible to define a new Pick List, it is possible to use the "Drop-down list box" control type to limit the choices just like a pick list.

First of all, when adding the data field, be sure to make it a "Text : General" field type.

When creating the Dialog definition you can use Quick Add Fields to add the new field to a dialog, which will add the input control as an "Edit text" control. Next you'll need to modify the control type to "Drop-down list box", and change the expressions appropriately.

The examples below show how the Initialize and OK/Save expressions should look -- of course you need to change "My List Input" to the proper control name, and use your real data field name in place of "<field name>".

The Initialize expression should look like:

```
Eval(DlgClearUserDropList("My List Input"), DlgAddUserDropListText("My List Input","Item 1"), DlgAddUserDropListText("My List Input","Item 2"),
DlgSetUserDropListSelectedText("My List Input", FieldText(This(),"<field name>")))
```

Of course you should replace "Item 1", etc. with the text you want to show, and you can add as many "Items" as you need by inserting more DlgAddUserDropListText functions (make sure DlgSetUserDropListSelectedText is still the last function, though).

The OK/Save expression should look like:

```
SetFieldText(This(),"<field name>", DlgGetUserDropListSelectedText("My List Input"))
```

Now the field will behave just like other pick-list based fields. The main difference is that changing the selections in the list by altering the expression above won't affect any previously entered data -- the old text will remain in the previous fields (resulting in the field appearing blank when viewed in the dialog again) -- so make sure the text selections are what you will always want to use, and avoid changing selections previously used (but they can be rearranged or added to later if needed).

Note: You'll need version 4.1 or later for the DlgClearUserDropList function -- without this, the list can keep duplicating itself if using Next/Prev buttons in dialogs like Customer Details or Site Details.

Removing information from receipts

We've had various questions about removing certain data from the standard receipt formats. This example shows how to make an alternate form that excludes the first night and last night information (this could be used for owners or seasonal renters that don't actually check in and out). A similar procedure can be used for removing most other information from receipts or forms.

What we need to do is import the sample "custom" version of the form we want to modify (the default ones in Forms Setup are simple add-ons to the canned formats so nothing can be removed from those). Then we'll need to modify the form, removing the Elements for the data we want to take off of the receipt.

The process is generally the same for most forms, though the exact location of the elements we need to remove may be different. This specific example will be for the "Full-page invoice" form.

1. Go to Maintenance / Advanced Customizations / Forms

2. Click the "Import form(s)" button.

3. The Import forms file dialog will open, and you need to navigate to the folder C:\Program Files\Campground Master\Samples. In most cases it will already be in C:\Program Files\Campground Master and you will just need to double-click the Samples folder to open it. If you don't see the Samples folder, you probably need to use the "Look In" drop-down list to navigate to the proper folder.

4. Locate the file "Sample Forms - Custom version of Full-page Invoice" (which may also have ".csv" at the end of it), click on it and then click Open.

5. Now at the bottom of the list you'll see "Custom Full-page Invoice". Double-click on it to edit that form (or click on it and click "Edit Form definition").

At this point it doesn't look like much -- there's only one Element in the list. But that's a "Region" element, which is actually a bunch of nested elements (this particular region just defines some outer borders for the rest of the form, so it's like a form within a form). There are a couple ways you could proceed from here. You could click "Expand Regions" so that all nested regions are broken out in a tree-like format, and then just find what you want. But that could be too confusing, so in this example we'll take it one step (level) at a time.

6. Double-click on the Region element to edit that element. Now a dialog will open showing the particulars of that region, including another list of elements (which are also mostly regions that further sub-divide the form into manageable and movable sections).

7. The dates we're looking for are in the upper right-hand area of the form (it's handy to have a printed version of the receipt to make it easier to follow along). So double-click on the region that has the Notes "Receipt #, Site/Resv info, Balance".

8. This of course opens up another list of elements... double-click on the Region with the Notes "Reservation & Site info" to edit that region.

9. Now we're down to mostly Text and Data Expression elements, though the parts we're interested happen to be more Regions. In this case the regions are just used to help format or position the information, so we don't need to drill down into those. We just want to remove all of the information related to First Night, Last Night, Check In and Check out -- so select all 8 of the elements that mention those in the Notes, and click "Delete Elements", then confirm the deletion.

10. That's basically it -- you'll need to click "Save" 3 times to un-stack the Edit Element dialogs for each Region that we drilled into, taking you back to the Edit Form Definition dialog.

11. You can change the Form name if you prefer to make it more appropriate (e.g. "Full page invoice, no dates"), then Save it.

12. If you want this new form to appear higher in the list when printing receipts, then you can select this form in the Forms Setup dialog and use "Move Up" to move it where you want it to appear in the list.

That's it for defining the new form. If you wanted to make this the new default for printing receipts, you could do so in Maintenance / Printing Options / Receipts.

Of course if you use several receipt formats then you would need to do this for each one you use. You could also delete or disable the original forms if you don't need any with the dates on them.

Color coding special reservation types

If you add special Reservation Types for weekly, seasonal or other special reservations (e.g. through the Pick Lists), then you may want to show them in different colors on the Rack. The normal Color Key only understands the standard "Base" reservation types, and can't be added to directly. However with the Color Schemes added in version 4.0, you can create your own color scheme for your new reservation types.

What you need to do is create a color scheme that has rules for each condition you want to check for (e.g. pending, checked in paid or not paid, etc.), for each new reservation type. We can use that color scheme to "override" the normal color scheme for reservations -- it will check your rules first, and if none of those conditions are met then it will use the standard color definitions (so you don't have to re-define all of the standard ones yourself). Of course you do need to be careful with your conditions so that it doesn't end up using one of your new colors when it shouldn't.

Here is an example of handling a new reservation type called "Weekly" (which of course would have to be added through Maintenance / Pick Lists / Reservation Types). The Base type for Weekly is assumed to be "Normal". Obviously this example can be altered for any reservation type by simply changing the text and/or numeric values in the rule expressions below.

1. Go to Maintenance / Advanced Customizations / Color Schemes.

2. Click "Add scheme definition", and give it a name like "Reservations override". Note that only one Color Scheme can be used to override the normal reservations colors, so if you have multiple types, etc. then the rules for all of them need to be defined in this one scheme.

3. Make sure the Default scheme is "Reservations" (this is what it uses after checking all of your rules).

4. Click "Add Color Rule", and give it a name. This is what will show in the Color Key for this color. Lets make this one "Reservation, Pending, Weekly, Paid" so it follows the format of the default color key labels.

5. Define the text and background colors using the "Change Text Color" and "Change Background Color" buttons.

6. Now you need to define the rule using an Expression, which must have a boolean result for the reservation we're checking. If it's True, then this color will be used. If it's False, then it will continue and check the next rule, and so forth.

This also means that the order of the rules may also be important in some cases (e.g. the highest priority conditions should be above the lower priority conditions in the rules list), but in this case it's not a factor since each case is very specific. Speed can also be affected by the order, e.g. the sooner it finds a match the quicker it finishes, but again it shouldn't matter for this simple case especially if Weekly is rare compared to Normal or other reservation types.

Now you could do all of the appropriate tests against the reservation to see if it's pending and see if it's paid in full as of the given date, etc. -- but there's a shortcut you can use. The function ColorStatusResv() will do the normal checks and return a code for which standard color key item it would normally use. All you need to do is see if this would be the color for a "Normal" type pending/paid reservation and then see if it's actually a Weekly reservation instead of Normal. By checking the Fixed List Values for "Site Status (color default)" you can find that the code for the "Resv, Pending, Paid" status is 35.

Note: If your reservation type has a different Base Type, e.g. Monthly instead of Normal, then of course this value would need to be changed accordingly.

So the expression for this condition would be:

```
ColorStatusResv(Resv(),ThisDate(),ThisPeriod()) = 35 AND Resv:Resv_Type =
"Weekly"
```

7. Save that rule.

8. Now we need to continue adding other rules to cover any other conditions like the normal color key does (if you skip some, then the reservation might not match any of your rules and thus be shown in the default "Normal" color).

You can save some time by selecting the last rule added above and "Copy Rule(s)", then edit the copied rule. Just change the name, colors, and the expression. Each expression should be the same as shown above except for the numeric code returned by ColorStatusResv(). The rest of the colors needed for a "Normal" based reservation type are shown below:

Reservation, Pending (not paid) : 28
Reservation, Pending, Don't Move : 30
Reservation, Pending, Don't Move, Paid : 37
Reservation, Confirmed : 25
Reservation, Confirmed, Don't Move : 51
Reservation, Guaranteed : 24
Reservation, Guaranteed, Don't Move : 50

Reservation, Checked In, Not Paid : 13
Reservation, Checked In, Paid : 14

Note that these numbers are only correct for Reservation Types with a "Normal" base type, and we don't test for Checked Out, Cancelled, etc. since those colors are the same for all reservation types anyway. There are some other special situations like Owner reservations (which don't have status so there's only paid/not paid status conditions to test), so be sure to check the "Site Status" fixed list values for all conditions you may need to test for the appropriate Base Type of your new reservation type.

9. Now once the scheme has all the rules and you Save it, there's one more thing you need to do -- make this scheme the default for reservations. Go to Maintenance / Advanced Customizations / Select Global Color Schemes, and select the scheme you just created in the drop-down list after "Override reservation names with color scheme:".

Now your new colors should be used on the Rack and other tab views, wherever a reservation name is normally colored. The Color Key will also show your new definitions along with the default ones, in case the operators need a reference.

Adding special reservation validations

This Advanced Customizations example is the most complex we've had so far, but the more adventurous of you may find it useful.

The goal here is to do some special validations when making a reservation, warn the operator if the validations fail, and give them the option to either continue or not. Since we need to do it *before* the reservation is completed, it must be done as a Dialog customization to the New Reservation dialog. We can accomplish what we need by adding elements of type "Action on data saved", and selecting the options in the elements to do the action before the normal processing and allow aborting the action.

Now there are some assumptions made here to simplify things -- we're assuming no linked reservations need to be checked, no complications with splitting reservations, etc... so this example isn't foolproof, but should cover most situations.

The three things we're going to validate are listed below. One is a common request to not allow continuing without "How found" filled in, and the other two are typical restrictions with some membership parks:

1. "How found" must be filled in.
2. At least a 7-day gap between reservations for the same customer.
3. No more than 3 "active" current or future reservations in place at any time.

Since this example involves a lot of data entry, it's not going to be a click-by-click tutorial. If you're with us this far, then hopefully you just need the expression details to fill in the blanks. Remember that when entering expressions, never use the Enter key -- while the text below may show multiple lines for an expression, that's just a limitation of this formatting -- keep entering the expression text as a single line (if using copy/paste, you may need to delete the extra line breaks manually).

Also note that while the first restriction is easy to test with a single expression, the other two involve a lengthy process of checking all other reservations for that customer. This is best done using a separate Script so the expression in the Elements are simplified. We've included the complete scripts below as well.

So, first we need to add a Dialog definition, with the following details:

<u>Dialog name:</u> New Reservation restrictions

<u>Add-on</u> (checked): New Reservation
<u>Enabled</u> (checked)

Note: Only one Add-on definition can be active for a given dialog, so if you already had such a definition for New Reservation then you would just add elements on to the existing one instead of creating a new one here.

Now we'll add the validations as 3 separate elements:

Element 1: (How found filled in)

Element Type: Action on data saved

Action expression:
```
IIFQ( Empty(DlgGetDropListSelectedText(1084)), 'Eval(MessageBox("How-found must be
filled in",0,2), DlgSetFocusCtrl(1084), .F.)', '.T.')
```

Execute the Action before... (checked)
Abort the operation which triggered... (checked)

Element 2: (Validate for 7-day gaps between reservations)

Element Type: Action on data saved

Action expression:
```
Eval(SetVar("pCust",ThisCust()),  SetVar("dFrom",DlgGetCtrlDate(1052)),
SetVar("dTo",DlgGetCtrlDate(1068)), iifq(B(CallScript("ResvGapUnder7")), '.T.',
'MessageBox("This would result in less than 7 days gap between reservations for this
customer.\\n\\nAre you sure you want to proceed?", 2, 1) = 2'))
```

Execute the Action before... (checked)
Abort the operation which triggered... (checked)

Element 3: (Validate for a max of 3 current/future reservations)

Element Type: Action on data saved

Action expression:
```
Eval(SetVar("pCust",ThisCust()), iifq(DlgGetCtrlDate(1068) < Today() OR
B(CallScript("ResvMax3Future")), '.T.', 'MessageBox("This would result in more
than 3 current or future reservations for this customer.\\n\\nAre you sure you
want to proceed?", 2, 1) = 2'))
```

Execute the Action before... (checked)
Abort the operation which triggered... (checked)

IMPORTANT: If you're using version 4.1 or earlier, then the last two Action expressions above must be modified -- instead of "\\n", you need to use "\n" (just one backslash instead of two, as explained in a previous article).

OK, that does it for the dialog definition, but before it will work, we need to add the following Scripts.

Script 1:

Script name: ResvGapUnder7

Description: Returns False if there's another reservation for this customer with less than a 7 day gap between them

Expressions: (The line numbers are not to be entered of course)

```
; Input:
;   dFrom and dTo vars should be the dates of the new reservation
;   pCust should be the customer this reservation is for
SetVar("fOK", .T.)
SetVar("nMinGap", 7)
SetVar("nList", CustNumResv(pCust))
SetVar( "n", 1)
WHILE n <= nList
  SetVar( "pResv", CustResvAt(pCust, n) )
  IF (dFrom > FieldDate(pResv,"Resv_Last_Date") AND Days(dFrom, FieldDate(pResv,"Resv_Last_Date")) < (nMinGap+1)) OR (dTo < FieldDate(pResv,"Resv_First_Date") AND Days(FieldDate(pResv,"Resv_First_Date"), dTo) < (nMinGap+1)) OR (dTo >= FieldDate(pResv,"Resv_First_Date") AND dTo <= FieldDate(pResv,"Resv_Last_Date")) OR  (dFrom >= FieldDate(pResv,"Resv_First_Date") AND dFrom <= FieldDate(pResv,"Resv_Last_Date")) OR (dFrom <= FieldDate(pResv,"Resv_First_Date") AND dTo >= FieldDate(pResv,"Resv_Last_Date"))
    IF ResvIsActive(pResv, .T. , .F. , .T., .T.) AND (FieldValue(ThisResv(),"Field_Index") != FieldValue(pResv,"Field_Index"))
      SetVar("fOK", .F.)
    ENDIF
  ENDIF
  SetVar("n", n+1)
ENDWHILE
; Returns False if there's another reservation for this customer with less than a 7 day gap between them
fOK
```

Script 2:

Script name: ResvMax3Future

Description: Returns False if this would make more than 3 future reservations

Expressions:

```
; Input:
;   pCust should be the customer this reservation is for
SetVar("nFuture", 0)
SetVar("nList", CustNumResv(pCust))
SetVar( "n", 1)
WHILE n <= nList
  SetVar( "pResv", CustResvAt(pCust, n) )
  IF FieldDate(pResv,"Resv_Last_Date") >= Today()
    IF ResvIsActive(pResv, .T. , .F. , .T., .T.) AND (FieldValue(ThisResv(),"Field_Index") != FieldValue(pResv,"Field_Index"))
      SetVar("nFuture", nFuture+1)
    ENDIF
  ENDIF
  SetVar("n", n+1)
ENDWHILE
nFuture < 3
```

That completes the necessary programming. With everything entered properly, it will perform the validations we needed. You'll notice that the two scripts are similar and basically perform a loop through all reservations of the given customer to check the necessary conditions. These examples could be modified to do other types of customer-based validation, such as looking for previous reservation balances, notes in previous reservations, types if sites they stayed in before, etc. You could also just do customer validation (e.g. without the loop), such as proper fields filled in, make sure the rig type is appropriate for the site selected, and so forth.