

How to Implement ServiceDesk's MyCriteria Option

This feature was added September of '09, and the first thing we want you to know is: There's a very strong chance you don't need it.

After all, hundreds of users performed in ServiceDesk very well, and over many years, without it. Please think of this feature as a potential bonus, something you may consider adding later, after doing other ServiceDesk setup and implementation. In other words, don't delay doing core ServiceDesk operations for the sake of learning about (and implementing) this feature.

Chapter 1

What is this Feature About?

Over the course of years, we've had requests from users for the addition of many different kinds of boxes, as directly connected to the general Callsheet/JobRecord structure. Some have wanted little radio/option buttons, for example, where they could click to indicate the type of payment expected. Others have wanted a checkbox where the call-taker would click, placing a check to provide explicit indication of having informed the caller of the service fee. Still others have wanted to attach complaint codes to each Callsheet (and resulting JobRecord). One client has even wanted to collect info about religious preference, to facilitate sending greeting cards on the occasion of religion-appropriate holidays.

For the most part, these pleas have been rebuffed, and for a couple of significant reasons: (1) limitations in regard to physical space as reasonably available for added controls and boxes within the existing forms; and (2) if we tried to explicitly add all the different kinds of controls and options that have been requested, we'd end up adding (likely) at least 50 new such controls. It would be boxes and buttons run amuck.

Instead, we figured it would make a lot more sense to give users the opportunity to make their own set of added controls — any particular set as might be wanted, to suit particular preference and need. In essence, you can make your own add-on information template.

That's what this feature is about.

Chapter 2

A Plea Against One Particular Purpose

At Rossware, we've been at this a long time. We've seen a large quantity and large variety of businesses implement our systems. Some things we see over and over.

Among these, it's very common for new users to believe they need an explicit (i.e., checkbox-type) designation, within the Callsheet/JobRecord interface, to indicate if the job is going to be COD, manufacturer's warranty, or other third-party billing. Obviously, ServiceDesk design does not provide that. Instead, the determination is implicit in how the party information is setup, and no separate, checkbox-type designation is needed.

But for new users, who are in a habit of having an explicit checkbox-type determination, the implicit one feels uncomfortable. Prior to beginning use, they often feel that having the explicit designation is going to be essential. The interesting thing is that, to date at least, after learning and using the implicit method *everyone* grows comfortable with it, and understands its virtue.

We describe the matter here because, if you're one of those new users who feels a strong, tugging need for an explicit designation, you may read about this *MyCriteria* feature and figure, "Aha, that's the solution!"

Ironically (if this is your situation), you will to some extent be correct. This *MyCriteria* feature could easily be used to create the very designation you're wanting.

Our whole purpose in this small chapter, however, is to plea with you (if, in fact, you want to pursue the *MyCriteria* feature for such a purpose) to resist. We think it would be a wasted effort — wasted, because if you'll just give yourself a chance, you'll soon find (like all others before you) that an explicit, separate-from-how-the-parties-are-setup designation is not something you truly need.¹

Please, don't use this feature for that purpose. Unless your situation is truly odd and extraordinary, you'll be better off otherwise.

¹ Another reason your effort would be wasted is because this *MyCriteria* information is not going to be printed on tickets; it's not going to be transferred to SD-Mobile, so on and etc. It will be available in the Callsheet/JobRecords themselves, and in at least one export. We hope you can understand: giving you the ability to use a custom template (and incorporate it with operational data) is — in and of itself — a major programming feat. To ask for such custom/user-defined data elements to translate into further and deeper elements of functionality is, simply, too much to ask.

Chapter 3

Creating Your *MyCriteria* Definition File

If ServiceDesk is going to create a set of controls, customized to your own design (i.e., and present to the user for input) — there is an obvious need for you to create and provide that design.

The method for doing so is via Excel (if you do not know, this is Microsoft's spreadsheet program, and is part of the Microsoft Office Suite).²

Begin by opening Excel and choosing to create a new document.

In that document, create headers in four columns, as follows:

	A	B	C	D
1	QuerySeries	ControlType	Label/Text	AnswerParameters
2				

The text in these headers will not have any substantive effect in ServiceDesk. You're putting the headers there, simply, to serve as handy reminders of the substance you're going to put under, in each column.

In that regard, you're going to provide one line-item for each "control" you want to have in your custom "MyCriteria" setup.

A control is a user-interface object, and in this context you can use any of four types (and as many³ or few of each as wanted):

- a Textbox;
- an Option Series;
- a CheckBox; or
- a Drop-Down List.

You should think carefully about the kinds of supplementary information you want to collect in your custom template, and about which kind of control will be optimum in regard to each.

Once you've figured the controls you want and what the associated labeling and text should be, it's time to type descriptive text, one line for each

² In early 2015 we added an option where you may use other than Excel for this purpose. You may use any application that produces a spreadsheet-type result, and save it in comma-delimited (.csv) format. Indeed, you could even open in NotePad, type each row with columns separated by simple commas, then save.

³ At this point the total quantity of controls (in whatever combination of types) should not exceed eight. We have not made space for more, than eight, in the one available data export. (Note that an *option series* counts as one control, no matter how many options are included within it.)

wanted control, in your Excel spreadsheet (please do not leave any blank lines). Here is an example of a possible result:

	A	B	C	D
1	QuerySeries	ControlType	Label/Text	AnswerParameters
2	1	CheckBox	I informed the customer of our rate structure	
3	1	CheckBox	Warned tech would not go if telephone not answered	
4	1	OptionSeries	How does this customer intend to pay	Cash Check CreditCard
5	1	TextBox	What is the customer's password	20
6	1	DropDownList	What is the FaultCode	FaultCodes.txt
7				

In the first column (QuerySeries), simply place the number "1". Don't worry more about contents for this column right now (it will be nothing more than a *later* concern, if ever, and is discussed in this Handbook's *Conclusion*).

In the second column, place text describing the kind of control you want to use. Please be sure, for any particular control, to use the exact ControlType text that you see in the examples. In other words, if you want to have a TextBox, be sure to precisely type the word "TextBox" (e.g., rather than "text box"). When ServiceDesk decodes what you provide, it's not going to employ human wisdom interpreting what was *meant*.

In the second column, place text the user needs to see, and as applicable to the query/control in question. In the case of a CheckBox, for example, it will be the text that appears *next to* a CheckBox. In the case of an Option Series, it will be the text on a frame surrounding your set of Button Choices. In the case of a TextBox, it will be the label immediately above the TextBox. In the case of a DropDownList, it will be the label immediately above the drop-down.

The final/fourth column is for answer parameters. It should be left blank if the control is a CheckBox.

For an Option Series control, however, it's used to indicate the series of buttons that should appear, and what the text should be on each. Simply provide one item of text for each wanted button, with each separated by a vertical bar (it's the upper-case character above your keyboard's Enter key).

For a TextBox control, the fourth column is used to indicate the maximum quantity of characters that will be allowed in the box. We recommend providing a number here that reasonably limits the user to no more than the maximum quantity of characters that should be needed. Please note, as well, this is not a suitable locale for essay-length insertions. The underlying data context is not structured to accommodate that.

For a DropDownList control, the fourth column is used to indicate a source for the content of your drop-down. That source will consist of a file which

lists each item as intended for the drop-down. In this column, please provide simply the file name (for details on the file itself, see the *Appendix* at end of this document).⁴

Once you have your MyCriteria template thus described, you simply need to save the spreadsheet in the location (and with the filename) where ServiceDesk expects to find it. Specifically, it must be saved as name **MyCriteria.xls**,⁵ and within the **x:\sd\netdata** folder on your server (where "x:" is whatever mapped letter you're using for the server).

Believe it or not, that's all the setup that's required. If you now restart ServiceDesk at any station (ServiceDesk only looks for this upon startup), it will find your new spreadsheet file, read the text out of out, and create a customized MyCriteria user-interface specifically per your design.

Chapter 4

Using in ServiceDesk

Happily, this does not require much instruction.

Prior to your creation of the *MyCriteria.xls* file, the *Criteria* button (as found on every Callsheet/JobRecord interface) was inactive:



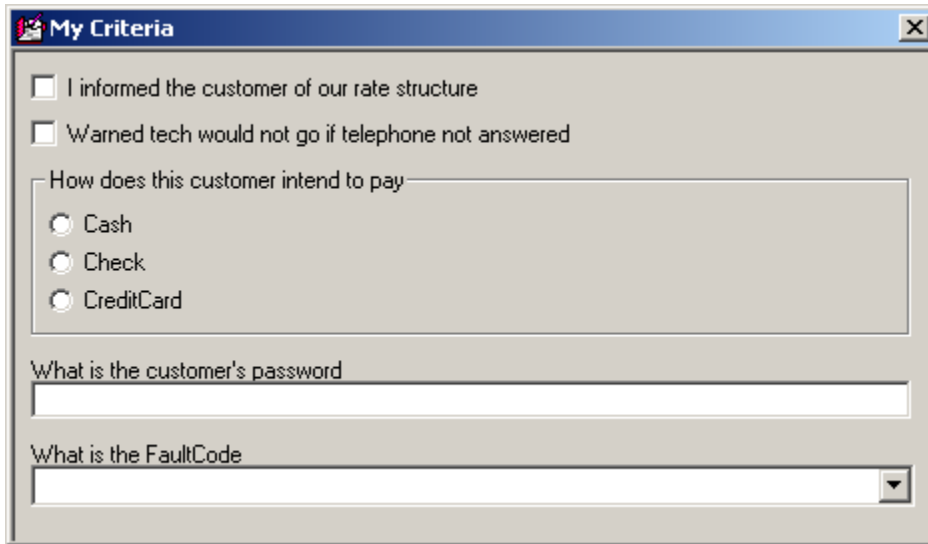
But now, when you start ServiceDesk and it sees your file, it will activate those buttons, because they now have a purpose.



⁴ Though in the text and illustration we've referred to just four columns, you may add a fifth if desired. You'd do so if wanting any of the MyCriteria values, as once inserted for a particular client, to be auto-repeated (on insertion of that customer's info), from a past job, into a new Callsheet. If so, make a fifth column. with "AutoRepeat" as the header. For each line-item where you want a past-input value to carry-through in the insertion process, place the word "Yes."

⁵ If you are electing to instead save as a comma-delimited file (i.e., .csv format) then instead select that as the desired format, and assure the filename extension is .csv rather than .xls.

More specifically, when you click on any such button, you'll instantly see your custom template, as specified by your *MyCriteria* file. In the case of our example spreadsheet, for example, the resulting template ends up looking like this:



Since it's a standard Windows interface, the user can click and fill-in just like in any similar context. And, like other windows in ServiceDesk, results are automatically saved.

Chapter 5

Enforcement

Upon first creating this feature (and prior to seeing even the first user try it), we're confident — since (as they say) “no good deed goes unpunished” — the result will be a demand for even more work. Specifically, some users will want us to program it so that a call-taker is forced to fill-in certain custom controls, before going on to other tasks.

The purpose of this small chapter is to advance-answer with an emphatic “No!” You have to do some of your own training and persuading, after all, and can't properly count on the software to exert all discipline.

Not that we can't at least help. In such regard, we've indeed provided a strong aid device. It consists of a color tag that lets you know, when glancing at a Callsheet or JobRecord, if all controls within the associated MyCriteria template have had a user response. Specifically, in any such case the *Criteria* button turns a muted blue color.



The notion is, if your call-takers are expected to fill-in your template in each instance, the presence of color or not provides ready indication of whether, in respect to any item, that task has been fulfilled.

One note in this regard: If a CheckBox control is clicked as checked, the act makes it evident the user has responded on that item. If it's simply left as unchecked, however (and even assuming it was *appropriately* left unchecked), it's not so evident the user has paid attention.

For this reason, we suggest, if there is a situation where it's legitimate for your call-taker to answer a query as either yes or no (i.e., it's not a situation where she should in all cases be checking to confirm she did something that should in all cases be done), instead of using a checkbox, use an Option Series consisting of a question, combined with "Yes" and "No" potential responses. That way, either a positive or negative response can easily be registered as such.

Chapter 6

Harvesting Your MyCriteria Data

In one respect, your MyCriteria data is immediately usable. Just open the Criteria sheet from any Callsheet or JobRecord, and you'll immediately see such information as any user has input.

We realize, however, that in all likelihood you'll want to make use of this data in other contexts. As of September 25, 2009, there is one other such locale. It's the *Scheduled Jobs Report Type 3*, available from the *Export Customer Data* form (Alt-F3). There are fields in that particular export to reflect data as created in conjunction with your MyCriteria setup. Just create the export, and you'll see them.

Chapter 7

Conclusion

We urge you to take great care in creating your MyCriteria setup — great care, in particular, in terms of structuring it in a manner that will well suit (and continue over time to well suit) your needs and preference.

The reason we urge this is simple. ServiceDesk will be saving data in conjunction with whatever structure you create. So long as that structure stays constant, the data will continue to fit the structure just fine (if you decide on occasion that you want to refine descriptive text, that's not significant in this regard; it's *structure* that counts). If you change the structure, all that data (as created when the structure was different) is not going to fit your structure any more. This could lead to all kinds of unpredictable (and unhappy) results.

This is the reason we placed that "QuerySeries" column in the MyCriteria spreadsheet structure. The thinking is, at some future point (assuming we're contacted by someone who feels the necessity for it), we may permit you to add a "Series 2" setup to your spreadsheet (which for then current usage would supersede Series 1, while still allowing old data to mate with Series 1). Or "Series 3," etc.

However, we've not yet created the ability to use further series, and strongly encourage you, if you're going to make a MyCriteria template at all, try hard to make it one that you can likely live with forever.

Appendix

Providing a Source File for any DropDown List

If you want a drop-down list, it will need to have list items (what good is a drop-down, otherwise?) This kind of thing is particularly apt, for example, in the case of complaint codes or fault codes.

The MyCriteria system is designed to expect a particular kind of file for use in populating drop-down lists (if such are used): specifically, a tab-delimited .txt file. You may create (or peruse and edit) the source, for any such file, in Excel. Or use any other suitable tool (technically, such a file can be created in any text editor). The important thing is to end up with a file of expected format (one line item for each that you want in the associated drop-down list, and with multiple columns, if any, separated by tabs).

Such files can be saved under whatever name you like, but as for location it must be the x:\sd\netdata folder on your server (where "x:" is whatever mapped letter you're using for the purpose). It's that folder where ServiceDesk will look for the particular FileName as specified in the fourth column (for any DropDownList item listing) of your MyCriteria.xls spreadsheet.

An example file (it provides Whirlpool fault-codes) is provided for you at this link: <http://rossware.net/MiniManuals/FaultCodes.txt>.

A final note about drop-downs: It's text from the first column only that will be saved in the ServiceDesk data that's directly connected to each Callsheet or JobRecord. Any further text (such as the description after a fault code, for example) is not part of that direct-saved data, and so will not be inherently included in a later export.