

Alt+Ed topic	Suggested re-wording
<p><b>ENT replaces ENT ...by ENT</b> Example of...</p> <p>Let's say we have a Department entity whose identity is always dependent on the company it's in. Further, if we delete the Company, we want all Department records for that Company deleted as well. Department is a perfect candidate to inherit from the Obsydian class Child.</p> <p>If I state that:</p> <p style="padding-left: 40px;"><b>Department is a Child</b></p> <p>I inherit the implicit "Owned by" triple:</p> <p style="padding-left: 40px;"><b>Department owned by Parent</b></p> <p>Since I want the parent of Department to be the Company, I would use the replacement and continuation triples:</p> <p style="padding-left: 40px;"><b>Department replaces Parent ...by Company</b></p> <p>...which will replace all occurrences of Parent with the Company entity I have defined in my model. Any processing associated with the template entity Parent will now be available as processing for the Department.</p>	<p><b>ENT replaces ENT ...by ENT</b> Example of...</p> <p>Suppose you have a Department entity whose identity always depends on the Company it's in. Further, if you delete a Company, you want all Department records for that Company deleted as well. Department is a perfect candidate to inherit from the Obsydian class Child.</p> <p>If you state that:</p> <p style="padding-left: 40px;">Department <b>is a</b> Child</p> <p>you inherit the implicit <b>owned by</b> triple:</p> <p style="padding-left: 40px;">Department <b>owned by</b> Parent</p> <p>where Parent is an abstract entity. Since you want the parent of Department to be Company, you can use the replacement triple:</p> <p style="padding-left: 40px;">Department <b>replaces</b> Company <b>...by</b> Company</p> <p>which replaces all occurrences of Parent with the Company entity you defined in your model. Any processing associated with the template entity Parent is now available as processing for the Department.</p>
<p><b>ENT replaces ENT ...by ENT</b> Try it...</p> <p style="text-align: center;">Following is a simulation of implementing</p> <p style="text-align: center;">ENT replaces ENT ...by ENT</p> <p>This practice uses the Obsydian Templates Editor to add both the "replaces" as well as the "...by" verbs at the same time. While you can add both of the triples manually, it is generally easier to do field replacement via templating.</p> <p style="text-align: center;">Click on the Next button to start the simulation.</p> <p style="text-align: right;"></p>	<p><b>ENT replaces ENT ...by ENT</b> Try it...</p> <p style="text-align: center;">Following is a simulation of implementing</p> <p style="text-align: center;">ENT <b>replaces</b> ENT <b>...by</b> ENT</p> <p>This practice uses the Obsydian Template Editor to add both the <b>replaces</b> triple and the <b>...by</b> continuation triple in one step. While you can add both of the triples manually, it is generally easier to do entity replacement via templating.</p> <p style="text-align: center;">Click Next to start the simulation.</p> <p style="text-align: right;"></p>
<p><b>ENT replaces FLD ...by FLD</b> What is it?</p> <p>The <b>ENT replaces FLD ...by FLD</b> triple and continuation verb allows you to replace a template field that was inherited from an abstract object definition with your own field definition.</p>	<p><b>ENT replaces FLD ...by FLD</b> What is it?</p> <p>The ENT <b>replaces FLD ...by</b> FLD triple enables you to replace a template field inherited from an abstract entity with a field you've defined.</p>

Alt+Ed topic	Suggested re-wording
<p><b>ENT replaces ENT ...by ENT</b> How to use</p> <p>You always use the <b>ENT replaces ENT</b> verb with its continuation verb ...by ENT. The target entity is a template that is replaced by the target of the continuation triple.</p> <p>There are two ways of entering this verb and it's continuation:</p> <ul style="list-style-type: none"> <li>• <b>You can enter this verb manually in the Obsydian Model Editor.</b> To do this: <ol style="list-style-type: none"> <li>1. First, add the appropriate ENT replaces ENT triple in the model editor, where the source object ENT is the entity you've defined, and the target object ENT is the target object of the "is a" inherited triple.</li> <li>2. Select the entire ENT replaces ENT triple you just added by clicking on the verb "replaces".</li> <li>3. Drag the entire triple to the source entry box of the Obsydian Model editor.</li> <li>4. Enter the continuation verb "by ENT", specifying the new Entity to use as the target object of the continuation verb.</li> <li>5. Don't forget to press enter or click the "+" sign to actually add the continuation triple.</li> </ol> </li> <li>• You can enter this verb <b>using the <a href="#">Template Editor</a></b> <ol style="list-style-type: none"> <li>1. Highlight the entire triple that was used to inherit the properties of whatever class you chose. To select the entire triple, click on the "is a" verb.</li> <li>2. Click on the Editor button.</li> <li>3. In the "Replaced by" column, type the name of the entity that will replace the template, abstract entity. You can also drag it from the Object Browser and drop it in the "Replaced by" field.</li> <li>4. Close the Template Editor window. When prompted whether you want to save the changes or not, make sure you click "Yes" (assuming that's what you want to do...).</li> </ol> </li> </ul> <p> <b>ObsyTip:</b> It's sometimes easier to use the Templates Editor to do replacement than trying to figure out what replaces what and which object gets replaced from which inherited triple. The Template Editor adds both the replacement and the continuation verb for you in one step.</p>	<p><b>ENT replaces ENT ...by ENT</b> How to use</p> <p>You always use an ENT <b>replaces</b> ENT triple with the continuation triple ...<b>by</b> ENT. The target entity is an abstract object that is replaced by the target of the continuation triple.</p> <p>You can enter this triple and its continuation in two ways:</p> <ul style="list-style-type: none"> <li>• In the Model Editor</li> <li>• Using the <a href="#">Template Editor</a></li> </ul> <p><b>To enter the triple using the Model Editor:</b></p> <ol style="list-style-type: none"> <li>1. In the Model Editor, add an ENT <b>replaces</b> ENT triple, where the source object is an entity you've defined, and the target object is the target of an <b>is a</b> inheritance triple.</li> <li>2. Select the ENT <b>replaces</b> ENT triple you just added, by clicking the verb (middle) column.</li> <li>3. Drag the entire triple to the source entry box of the Model Editor.</li> <li>4. Select the continuation verb ...<b>by</b> ENT, and specify your new entity as the target object.</li> <li>5. Press ENTER or click the Add button  to add the continuation triple.</li> </ol> <p><b>To add the triple using the Template Editor:</b></p> <ol style="list-style-type: none"> <li>1. In the Model Editor, highlight the entire <b>is a</b> inheritance triple. To select the entire triple, click on the <b>is a</b> verb (the middle column).</li> <li>2. Click the Editor button .</li> <li>3. In the Replaced By column, type the name of the entity that will replace the abstract entity. You can also drag it from the Object Browser.</li> <li>4. Close the Template Editor and click Yes to save your changes.</li> </ol> <p> <b>ObsyTip:</b> It's sometimes easier to use the Template Editor to do replacement than trying to figure out what replaces what and which object gets replaced from which inherited triple. The Template Editor adds both the <b>replaces</b> triple and the continuation triple for you in one step.</p>
<p><b>ENT replaces FLD ...by FLD</b> Why use it?</p> <p>Once again, the reason you should use any type of replacement boils down to the fact that you cannot change an inherited triple. So, if you want to customize a field that has been inherited as a result of an "is a ENT" verb, the only way to do that is using the ENT replaces FLD ...by FLD triple.</p>	<p><b>ENT replaces FLD ...by FLD</b> Why use it?</p> <p>The reason for using any type of replacement is that you cannot change an inherited triple. If you want to customize a field that has been inherited as the result of an ENT <b>is a</b> ENT triple, you must use an ENT <b>replaces</b> FLD ...<b>by</b> FLD triple.</p>

Alt+Ed topic	Suggested re-wording
<p><b>ENT replaces FLD</b> Example of</p> <p>Let's say that I want the key of one of the files in my model – Student – to be generated automatically each time a new record is added. The Obsydian class library that gives me this functionality is Enumerated Entity.</p> <p>When I say that</p> <ul style="list-style-type: none"> <li>• <b>Student is a Enumerated Entity</b></li> </ul> <p>one of the triples that is inherited from Enumerated Entity is:</p> <ul style="list-style-type: none"> <li>• <b>Enumerated Entity known by Transaction Number.</b></li> </ul> <p>This results in an implicit triple of</p> <ul style="list-style-type: none"> <li>• <b>Student known by Transaction number.</b></li> </ul> <p>Since I want the key of my file to be Student ID – not Transaction number, but I can't change the inherited triples, I'll use replacement to override all occurrences of Transaction number with Student ID. In order to avoid domain problems, I'll say that</p> <ul style="list-style-type: none"> <li>• <b>Student ID is a Transaction number</b></li> </ul> <p>And then replace Transaction number with Student ID, resulting in the triple:</p> <ul style="list-style-type: none"> <li>• Student replaces Transaction number ...by Student ID.</li> </ul> <p>Now, anywhere Student would have used Transaction number – in a view, panel design, action diagram statement, parameter list (or anywhere, for that matter), Obsydian will use Student ID instead.</p>	<p><b>ENT replaces FLD ...by FLD</b> Example of...</p> <p>Suppose you want the key of one of the entities in your model – Student – to be generated automatically each time a new record is added. You can use the Obsydian class library object OBASE/Enumerated Entity to do this.</p> <p>When you say that:</p> <p style="padding-left: 40px;">Student <b>is a</b> Enumerated Entity</p> <p>you inherit the following triple:</p> <p style="padding-left: 40px;">Enumerated Entity <b>known by</b> Transaction Number</p> <p>This results in the implicit triple:</p> <p style="padding-left: 40px;">Student <b>known by</b> Transaction Number</p> <p>Since you want the key of your entity to be Student ID – rather than the inherited Transaction Number – and you can't change inherited triples, you can use replacement to override all occurrences of Transaction Number with Student ID. To avoid domain problems, assume that:</p> <p style="padding-left: 40px;">Student ID <b>is a</b> Transaction Number</p> <p>And then replace Transaction Number with Student ID, resulting in the triple:</p> <p style="padding-left: 40px;">Student <b>replaces</b> Transaction Number ...by Student ID</p> <p>Now, anywhere Student would have used the inherited key Transaction Number – for example, in a view, panel design, action diagram statement, or parameter list – Obsydian will use Student ID instead.</p>
<p><b>Demo of ENT replaces FLD (no title actually shown)</b></p> <p>While we could have added each of the triples manually, by using the Entity Templates editor Obsydian adds both the "replaces" triple as well as the "...by" verb at the same time.</p> <p>Click the restart button to rerun this demo, or the exit button to stop it.</p> <div style="text-align: center;">  </div>	<p><b>ENT replaces FLD ...by FLD</b> Demonstration</p> <p>While you could add each of the triples manually, when you use the Template Editor, Obsydian adds both the <b>replaces</b> triple and its <b>...by</b> continuation triple in a single step.</p> <p>Click the restart button to rerun this demo, or the exit button to stop it.</p> <div style="text-align: center;">  </div>

Alt+Ed topic	Suggested re-wording
<p><b>ENT replaces FLD ...by FLD</b> How to use it</p> <p>You always use the <b>ENT replaces FLD</b> with its continuation verb ...by FLD. The target field is a template that is replaced by the field specified by the continuation triple.</p> <p>There are two ways of entering the <b>ENT replaces FLD</b> verb and it's <b>...by FLD</b> continuation:</p> <ul style="list-style-type: none"> <li><b>You can enter it manually in the Obsydian Model Editor.</b> To do this: <ol style="list-style-type: none"> <li>First, add the appropriate ENT replaces FLD triple in the model editor, where the source object ENT is the entity you've defined, and the target object FLD is the field inherited as a result of the "is a" triple.</li> <li>Select the entire ENT replaces FLD triple you just added by clicking on the verb "replaces".</li> <li>Drag the entire triple to the source entry box of the Obsydian Model editor.</li> <li>Enter the continuation verb "by FLD", specifying the new Field to use as the target object of the continuation verb.</li> <li>Don't forget to press enter or click the "+" sign to actually add the continuation triple.</li> <li>Note that it is a good idea to have your new field inherit from the template field you are replacing, in order to avoid any domain problems down the line.</li> </ol> </li> <li>You can also enter this verb <b>using the <a href="#">Template Editor</a></b> <ol style="list-style-type: none"> <li>Highlight the entire triple that was used to inherit the properties of whatever class you chose. To select the entire triple, click on the "is a" verb.</li> <li>Click on the Editor button.</li> <li>In the "Replaced by" column, type the name of the field that will replace the template field. You can also drag the field from the Object Browser and drop it in the "Replaced by" field.</li> <li>Close the Template Editor window. When prompted whether you want to save the changes or not, make sure you click "Yes" (assuming that's what you want to do...).</li> </ol> </li> </ul> <p><b>ObsyTip:</b> It's sometimes easier to use the Templates Editor to do replacement than trying to figure out what replaces what and which object gets replaced from which inherited triple. The Template Editor adds both the replacement and the continuation verb for you in one step.</p>	<p><b>ENT replaces FLD ...by FLD</b> How to use</p> <p>You always use an <b>ENT replaces FLD</b> triple with the continuation triple ...by FLD. The target field is an abstract field that is replaced by the target of the continuation triple.</p> <p>You can enter this triple and its continuation in two ways:</p> <ul style="list-style-type: none"> <li>In the Model Editor</li> <li>Using the <a href="#">Template Editor</a></li> </ul> <p><b>To the triple using Model Editor:</b></p> <ol style="list-style-type: none"> <li>In the Model Editor, add an ENT <b>replaces</b> FLD triple, where the source object is the entity you've defined, and the target object is a field you inherited as a result of an <b>is a</b> triple.</li> <li>Select the ENT <b>replaces</b> FLD triple you just added, by clicking the verb (middle) column.</li> <li>Drag the entire triple to the source entry box of the Model Editor.</li> <li>Select the continuation verb <b>...by FLD</b>, and specify your key field as the target object.</li> <li>Press ENTER or click the Add button  to add the continuation triple. <p><b>Note:</b> To avoid domain problems, it is a good idea for your new field to inherit from the template field you are replacing.</p> </li> </ol> <p><b>To add the triple using the Template Editor:</b></p> <ol style="list-style-type: none"> <li>In the Model Editor, highlight the entire <b>is a</b> inheritance triple. To select the entire triple, click on the <b>is a</b> verb (the middle column).</li> <li>Click the Editor button .</li> <li>In the Replaced By column, type the name of the field that will replace the template field. You can also drag the field from the Object Browser.</li> <li>Close the Template Editor and click Yes to save your changes.</li> </ol> <p> <b>ObsyTip:</b> It's sometimes easier to use the Template Editor to do replacement than trying to figure out what replaces what and which object gets replaced from which inherited triple. The Template Editor adds both the <b>replaces</b> triple and the continuation triple for you in one step.</p>

Alt+Ed topic	Suggested re-wording
<p><b>ENT replaces FLD ...by FLD</b> Try it...</p> <p>Following is a simulation of implementing</p> <p style="text-align: center;">ENT replaces FLD ...by FLD</p> <p style="text-align: center;">In Obsydian.</p> <p>This practice uses the Obsydian Templates Editor to add both the "replaces" as well as the "...by' verbs at the same time.</p> <p>While you can add both of the triples manually, it is generally easier to do a field replacement via templating.</p> <p>Click on the "Next" button to start the simulation</p> <div style="text-align: right;">  </div>	<p><b>ENT replaces FLD ...by FLD</b> Try it...</p> <p>Following is a simulation of implementing</p> <p style="text-align: center;"><b>ENT replaces FLD</b> <b>...by FLD</b></p> <p>This practice uses the Obsydian Template Editor to add both the <b>replaces</b> triple and the <b>...by</b> continuation triple in one step. While you can add both of the triples manually, it is generally easier to do field replacement via templating.</p> <p>Click Next to start the simulation</p> <div style="text-align: right;">  </div>
<p><b>FNC replaces VW ...by VW</b> <b>How to Implement</b></p> <p>You always use this verb with its continuation verb ...by VW. The target view is replaced by the view specified by the continuation triple.</p> <p>To perform the replacement:</p> <ol style="list-style-type: none"> <li>1. First, add the appropriate FNC replaces VW triple in the model editor. The FNC you will use as the source object is the function for which you wish to replace the view. The VW target object is the view which was inherited from the "is a" triple that you wish to replace.</li> <li>2. Select the entire FNC replaces VW triple you just added by clicking on the verb "replaces".</li> <li>3. Drag the entire triple to the source entry box of the Obsydian Model editor.</li> <li>4. You can then enter the continuation verb "by VW", specifying the new view upon which you want to base the function as the object in the target entry box.</li> <li>5. Don't forget to press enter or click the "+" sign to actually add the continuation verb.</li> </ol>	<p><b>FNC replaces VW ...by VW</b> <b>How to use</b></p> <p>You always use a FNC <b>replaces</b> VW triple with the continuation triple <b>...by</b> VW. The target view is an abstract object that is replaced by the target of the continuation triple.</p> <p><b>To perform the replacement:</b></p> <ol style="list-style-type: none"> <li>1. In the Model Editor, add a FNC <b>replaces</b> VW triple, where the source object is the function whose view you want to replace, and the target object is the target of an <b>is a</b> inheritance triple.</li> <li>2. Select the entire FNC <b>replaces</b> VW triple you just added, by clicking the verb (middle) column.</li> <li>3. Drag the entire triple to the source entry box of the Model Editor.</li> <li>4. Select the continuation verb <b>...by</b> VW, and specify the new view on which to base your function as the target object.</li> <li>5. Press ENTER or click the Add button  to add the continuation triple.</li> </ol>

Alt+Ed topic	Suggested re-wording
<p><b>FNC replaces FNC ...by FNC</b> Why use it?</p> <p>In Obsydian, you will often inherit calls to functions by way of the "is a" triple. These calls can originate from either Obsydian class library objects, or from other functions you have designed in Obsydian and used as the target of an "is a" verb.</p> <p><b>FNC replaces FNC ....by FNC</b> is used frequently to force a reference to a specified function to be templated. Note that using this verb is referred to as manual templating; that is, you are explicitly making the replacement to a function which you specify. In contrast, <a href="#">automatic templating</a> occurs as a result of implicit inheritance on a scoped object when you inherit functions from a class library.</p> <p>For a function that inherits references to another function, this verb lets you replace those references with another function. The replacement applies to all inherited triples and action diagram statements that reference the original function.</p>	<p><b>FNC replaces FNC ...by FNC</b> Why use it?</p> <p>In Obsydian, you often inherit calls to functions by way of an <b>is a</b> triple. These functions can originate from Obsydian class library objects or from functions you have designed in Obsydian and used as the target of an <b>is a</b> triple.</p> <p><b>FNC replaces FNC ...by FNC</b> is used frequently to force a reference to a specified function to be templated. This is referred to as <i>manual templating</i>; that is, you are explicitly making the replacement to a function you specify. In contrast, <a href="#">automatic templating</a> occurs as a result of implicit inheritance on a scoped object.</p> <p>For a function that inherits references to another function, you can use this triple to replace those references with another function. The replacement applies to all inherited triples and action diagram statements that reference the original function.</p>
<p><b>FNC replaces FNC ...by FNC</b> Example of</p> <p>Suppose I have a Customer maintenance function that inherited from User Maintained Entity. The Maintenance Suite.Maintenance UI includes calls to the Create, Change and Delete user interface panels. Therefore, the Customer.Maintenance UI will include calls to the Customer.Maintenance Suite.Create user interface, Customer.Maintenance Suite.Change user interface and Customer.Maintenance Suite.Delete user interface.</p> <p>I've decided that I want to replace the Change user interface function with some other type of function – a Tab Dialog Edit, perhaps.</p> <p>Again, since the function calls were inherited, I cannot change them directly. However, I could use the</p> <p>FNC replaces FNC ...by FNC to substitute the call to the Tab Dialog Edit where the original call to the Change User Interface occurred.</p> <p>To do this, all I would have to do is enter:</p> <ul style="list-style-type: none"> <li>• <b>...Maintenance UI replaces Change User Interface ...by Tab Dialog Edit</b></li> </ul> <p>Anywhere the Maintenance UI function references the Change User Interface in triples or action diagram statements, Obsydian will instead replace it with the Tab Dialog Edit function.</p>	<p><b>FNC replaces FNC ...by FNC</b> Example of...</p> <p>Suppose you have a Customer Maintenance function that inherits from OBASE/User Maintained Entity. The function Maintenance Suite.Maintenance UI calls the functions Create User Interface, Change User Interface, and Delete User Interface. Therefore, your Customer.Maintenance Suite calls the corresponding inherited user interface functions.</p> <p>Suppose also that your Customer entity inherits from OBASE/Entity With Tab Dialog and you want to replace the Change User Interface function with the inherited Tab Dialog Edit function.</p> <p>Since the function calls were inherited, you cannot change them directly. However, you can use a <b>FNC replaces FNC ...by FNC</b> triple to substitute a call to Tab Dialog Edit for the original call to Change User Interface.</p> <p>To do this, enter the following:</p> <p>Customer.Maintenance Suite.Maintenance UI <b>replaces</b> Change User Interface <b>...by</b> Tab Dialog Edit</p> <p>Anywhere the Maintenance UI function references Change User Interface in triples or action diagram statements, Obsydian replaces it with your Tab Dialog Edit function.</p>

Alt+Ed topic	Suggested re-wording
<p><b>FNC replaces FNC ...by FNC</b> How to Implement</p> <p>You always use the <b>FNC replaces FNC</b> triple with its continuation verb <b>...by FNC</b>. The target function is replaced by the function specified by the continuation triple.</p> <p>To perform the replacement:</p> <ol style="list-style-type: none"> <li>1. First, <b>add the appropriate FNC replaces FNC triple</b> in the model editor. The FNC you will use as the source object is the function that will call the function you wish to replace. The target FNC object is the function you want to replace that was inherited from the "is a" triple on the source object function.</li> <li>2. <b>Select the entire FNC replaces FNC triple</b> you just added by clicking on the verb "replaces".</li> <li>3. <b>Drag the entire triple to the source entry box</b> of the Obsydian Model editor.</li> <li>4. You can then <b>enter the continuation verb "by FNC", specifying the new function</b> upon which you want to base the function as the object in the target entry box.</li> <li>5. Don't forget to press enter or click the "+" sign to actually add the continuation verb.</li> </ol>	<p><b>FNC replaces FNC ...by FNC</b> How to use</p> <p>You always use a <b>FNC replaces FNC</b> triple with its continuation verb <b>...by FNC</b>. The target function is an abstract object that is replaced by the target object of the continuation triple.</p> <p><b>To perform the replacement:</b></p> <ol style="list-style-type: none"> <li>1. In the Model Editor, add a <b>FNC replaces FNC</b> triple, where the source object is the function that calls the function you want to replace and the target object is target of an <b>is a</b> inheritance triple.</li> <li>2. Select the entire <b>FNC replaces FNC</b> triple you just added, by clicking on the verb (middle) column.</li> <li>3. Drag the entire triple to the source entry box of the Model Editor.</li> <li>4. Select the continuation verb <b>...by FNC</b>, and specify your new function as the target object.</li> <li>5. Press ENTER or click the Add button  to add the continuation triple.</li> </ol>
<p><b>FLD replaces FLD ...by FLD</b> What is it?</p> <p>The <b>FLD replaces FLD ...by FLD</b> triple lets you replace an inherited version of a field with another field that is <a href="#">scoped</a> to the same object. Generally, this will only happen as a result of inheriting from a <a href="#">structured</a> field or a field that contains processing associated with it (e.g. a "<a href="#">has derived</a>" or a "<a href="#">computed by</a>" function).</p> <p>Compared with some of the other replacement triples, this triple is used less often and only in special circumstances.</p>	<p><b>FLD replaces FLD ...by FLD</b> What is it?</p> <p>The <b>FLD replaces FLD ...by FLD</b> triple lets you replace an inherited version of a field with another field that is <a href="#">scoped</a> to the same object. Generally, this only happens when you inherit from a <a href="#">structured</a> field or a field that contains processing associated with it (such as a <a href="#">has derived</a> or <a href="#">computed by</a> function).</p> <p>This triple is used less often than the other replacement triples, and only in special circumstances.</p>
<p><b>FLD replaces FLD ...by FLD</b> Why use it?</p> <p>You would use <b>FLD replaces FLD ...by FLD</b> whenever you have inherited from a field that has field(s) scoped to it and you want to replace the inherited fields with other fields of your own. Two instances when this might happen is when you have fields that are included in a structured field (e.g. Address includes Address Line 1, Address Line 2, Zip Code, etc.), or a field that has a function associated with it and contains parameters which might need to be changed based on the usage of the field.</p>	<p><b>FLD replaces FLD ...by FLD</b> Why use it?</p> <p>Use <b>FLD replaces FLD ...by FLD</b> when you have inherited from a field that has one or more fields scoped to it and you want to replace an inherited field with one of your own.</p> <p>You might want to use this:</p> <ul style="list-style-type: none"> <li>• For fields that are included in a structured field (such as an Address field that includes the fields Address Line 1, Address Line 2, and Zip Code)</li> <li>• For a field that has a function associated with it and contains parameters that might need to be changed based on how you use the field</li> </ul>

Alt+Ed topic	Suggested re-wording
<p><b>FLD replaces FLD ...by FLD</b> Example</p> <p>Let's say we have a <a href="#">structured field</a> called (Address) which is made up of several fields. [BTW, the parentheses indicate a naming convention for a structure field]. In the Model Editor I would expect to see some triples along the following lines:</p> <p><b>(Address) includes FLD Address Line 1</b>  <b>(Address) includes FLD Address Line 2</b>  <b>(Address) includes FLD City</b>  <b>(Address) includes FLD State</b>  <b>(Address) includes FLD Zip Code</b></p> <p>Since the FLD Includes FLD triple is a scoping triple, each of the resulting fields is scoped to (Address).</p> <p>Then if I say that:</p> <p><b>Customer has FLD Customer Address</b></p> <p>and that</p> <p><b>Customer address is a FLD (Address)</b></p> <p>this results in Customer having all of the fields in the structured field (Address).</p> <p>Now let's say that we need a specialized version of the Customer entity to keep track of our European customers. For the most part, the characteristics and functionality I need for European Customer is the same as my standard Customer entity. Therefore I would say that:</p> <p><b>Euro Customer is a ENT Customer</b></p> <p>However, since my European Customers have a Postal Code in their address rather than a Zip Code, I will define a Postal code field scoped to the European Address field by adding the triple:</p> <p><b>Euro Customer includes FLD Postal Code</b></p> <p>Now I can create a new structured field for these addresses by saying that:</p> <p><b>Euro Address is a FLD (Address)</b></p> <p>and then replace any occurrences of Zip Code with Postal Code by using the FLD replaces FLD ...by FLD triple:</p> <p><b>Euro Address replaces FLD Euro Address.Zip code ...by FLD Euro Address.Postal Code</b></p> <p>Note that in performing the replacement, the field being replaced and the replacing field must be scoped to the same object. By using the "Includes FLD" triple to define the new Postal Code field as being part of Euro Address, this resulted in the scoped field "Euro Address.Postal Code". If I had defined Postal Code as a non-scoped field, the replacement would not have worked correctly. In this case, I could also re-scope the Postal Code field by dragging it and dropping it onto the Euro Address field in the Object Browser.</p>	<p><b>FLD replaces FLD ...by FLD</b> Example of...</p> <p>Suppose you have a <a href="#">structured field</a> called (Address) that contains several fields. The definition of the (Address) field could be something like the following, where the parentheses around the field name indicate a structured field:</p> <p>(Address) <b>includes</b> Address Line 1  (Access) <b>includes</b> Address Line 2  (Address) <b>includes</b> City  (Address) <b>includes</b> State  (Address) <b>includes</b> Zip Code</p> <p>Since the FLD <b>includes</b> FLD triple is a scoping triple, each of the resulting fields is scoped to (Address).</p> <p>If you say that:</p> <p>Customer <b>has</b> Customer Address</p> <p>and</p> <p>Customer Address <b>is a</b> (Address)</p> <p>then Customer Address has all of the fields in the structured field (Address).</p> <p>Now suppose you need a specialized version of the Customer entity to keep track of European Customers. For the most part, the characteristics and functionality you'll need to track European Customers is the same as that of the standard Customer entity. Therefore you can start with:</p> <p>Euro Customer <b>is a</b> Customer</p> <p>However, since European Customers have a Postal Code in their address rather than a Zip Code, you can define a Postal Code field scoped to the European Address field by adding the triple:</p> <p>Euro Address <b>includes</b> Postal Code</p> <p>Now you can create a new structured field for these addresses by saying that:</p> <p>Euro Address <b>is a</b> (Address)</p> <p>and then replace any occurrences of Zip Code with Postal Code by using the FLD <b>replaces</b> FLD ...by FLD triple:</p> <p>Euro Address <b>replaces</b> Euro Address.Zip code ...by Euro Address.Postal Code</p> <p>For this replacement, the field being replaced and the replacing field must be scoped to the same object. Using a FLD <b>includes</b> FLD triple to define the new Postal Code field as part of Euro Address created the scoped field Euro Address.Postal Code. If you had defined Postal Code as an unscoped field, the replacement would not have worked correctly. In this case, you could also re-scope the Postal Code field by dragging it and dropping it onto the Euro Address field in the Object Browser.</p>

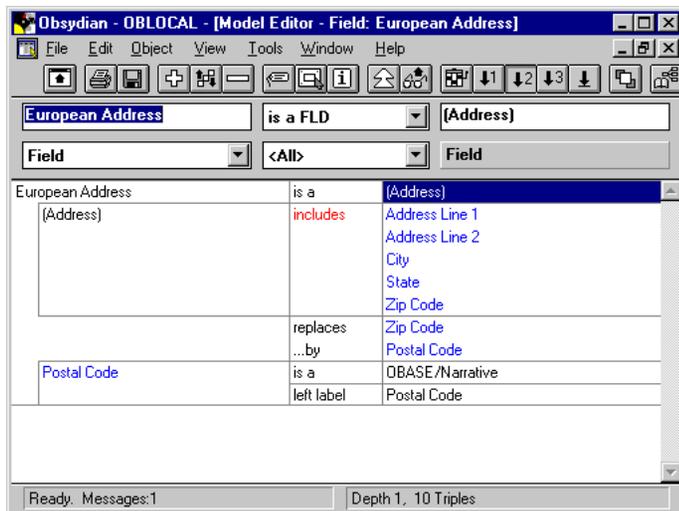
**Alt+Ed topic****FLD replaces FLD ...by FLD****How to Implement**

You always use the **FLD replaces FLD** with its continuation verb **...by FLD**. The target field is a template that is replaced by the field specified by the continuation triple. Both the replaced and the replacing field must be at the same scoping level.

To enter the **FLD replaces FLD** verb and it's **...by FLD** continuation:

1. First, **add the appropriate FLD replaces FLD triple** in the model editor, where the source object FLD is the field you've defined somewhere else with an "is a FLD" triple, and the target object FLD is the field inherited as a result of that "is a" triple.
2. **Select the entire FLD replaces FLD triple** you just added by clicking on the verb "replaces".
3. **Drag the entire triple to the source entry box** of the Obsydian Model editor.
4. **Enter the continuation verb "by FLD", specifying the new Field to use as the target object** of the continuation verb. Note that the field you enter as the target of the continuation verb "...by" must be at the same scoping level as the field you are replacing (e.g. the same number of "dots" in the name). In order to make sure both fields are in the same domain, the replacing field should inherit from the replaced field.

The resulting triples might look something like this:

**Suggested re-wording****FLD replaces FLD ...by FLD****How to use**

You always use a **FLD replaces FLD** triple with its continuation triple **...by FLD**. The target field is a template field that is replaced by the target of the continuation triple. Both the source and the target fields must be at the same scoping level.

**To perform the replacement:**

1. In the Model Editor, add a **FLD replaces FLD** triple, where the source object is a field you've defined in your model, and the target object is the target of an inheritance triple.
2. Select the entire **FLD replaces FLD** triple you just added by clicking the verb (middle) column.
3. Drag the entire triple to the source entry box of the Model Editor.
4. Select the continuation verb **...by FLD**, and specify your new field as the target object.

**Note:** The field you enter as the target of the continuation triple must be at the same scoping level as the field you are replacing. In other words, it must have the same number of "dots" in the name. To ensure that both fields are in the same domain, the replacing field should inherit from the replaced field.

The resulting triples might look something like this:

