



Knowledgeone^{K1}

MODIFYING THE KNOWLEDGEONE^{K1} DATA MODEL

Modifying the Knowledgeone^{K1} Data Model

Introduction	3
Overview	3
**Warning	3
Step One - Make Sure it Doesn't Already Exist!	4
Normalized, Relational Design – Take Care	4
Existing 'Out-of-the-box' Personalities.....	4
How Do You Know Which Tables To Select For Each Personality?.....	5
Planning Your Security Regime	6
Overview.....	6
Why Are Security Groups Necessary?.....	6
Why Do You Need Field Level Security?	6
Why Do You Need Security Codes?.....	6
Changes Via The DRM.....	7
Overview.....	7
Detail – Knowledgeone ^{K1} Help Screens.....	7
Tips On Using The DRM.....	8
Changes Via The UI	9
Overview.....	9
Detail – Knowledgeone ^{K1} Help Screens.....	9
Making Changes To Knowledgeone ^{K1} Via The UI.....	9
Make Fieldlinks Visible Or Invisible – Seeing If A Fieldlink Already Exists.....	9
Create Auto Number Formats	13
Create Autonumber Format Multiplesequences	15
Add Filters.....	15
Make Fields Type Code Dependent.....	16
Make Fieldlinks Type Code Dependent	17
Adding Forms to Knowledgeone ^{K1}	18
Overview.....	18
Data-Entry Forms	18
Data-Output Forms.....	19
Active Reports.....	20
Tables and Methods	20
“I have added a new table but now I can't Modify it. Why?”	20
Data Dictionary.....	21
“How do I find out what tables and fields are in Knowledgeone ^{K1} ?”	21

Introduction

Having just delivered a whole new genre of application software, a "Generic Application Solution" (GAS), where the customer can change both the data model and the processes, we are now getting requests about "how to?". This paper is about changing the Data Model.



Note: This paper does not attempt to duplicate the information in the Knowledgeone^{K1} help screens; it is designed to complement that information.

The help screens in Knowledgeone^{K1} are a superset of normal help screens and have been designed to act both as a user guide and a developer's guide. For example, refer to the topic "Changing K1 – Q&A". You will find that there is a wealth of information within the help system. And, the added advantage is that it is always online when you are working with Knowledgeone^{K1} or the DRM; no searching for misplaced manuals is required.

However, there can never be too much information so we have decided to produce this paper plus a series of online demos on all aspects of Knowledgeone^{K1}, (the first, an Introduction to Knowledgeone^{K1}, is on the website now).

This paper is intended to demystify the 'Changing Knowledgeone^{K1}' process and present it in a 'paint-by-numbers' fashion. It is also intended as an expansion of the help screens. Throughout this paper, I will continually refer to the help screens rather than duplicate information within this paper. And, if you begin to get the idea that I am trying to incentivize you to read the help screens then you are correct.

Overview

1. The Knowledgeone^{K1} data model is able to be changed by each customer.
2. The tool provided by Knowledgeone Corporation to accomplish this task is called the DRM, the Database Registration Module.
3. All Knowledgeone^{K1} Table and Field Additions, Modifications and Deletions ****MUST** be done via the DRM.
4. Other changes to the visible data model (e.g., making FieldLinks visible/invisible, adding Filters and configuring tables by Type Code) **MUST** be made using the Knowledgeone^{K1} User Interface, (UI).
5. You **MUST** be defined as a Knowledgeone^{K1} Administrator with access to all Tables and all Methods to be able to make these changes.

If you are given access to the DRM but do not have access to all tables and methods you will experience problems when trying to make changes.

****Warning**

If any Knowledgeone^{K1} Table and Field Additions, Modifications and Deletions are done via any other processor (e.g., SQL Enterprise Manager) database integrity will be lost and Knowledgeone^{K1} will not work correctly.

Step One - Make Sure it Doesn't Already Exist!

Before making any changes to the Knowledgeone^{K1} Data Model, **PLEASE** make sure that they are necessary.

By design, Knowledgeone^{K1} includes many tables and fields and fieldLinks that are either underutilised or 'not turned on' or 'invisible'. We have tried to allow for most expected Data Model extensions by including more tables and fields and links than are required for the standard 'out-of-the-box' Personalities of Knowledgeone^{K1}. The standard Knowledgeone^{K1} Data Model should in fact be a superset for most customers.

Make sure you first look at the Table table, the Field Table and the FieldLink table BEFORE deciding to create a new table or new field or new fieldLink.

For example, if you decide to extend the CRM Personality by importing details of customer purchase orders and invoices, you will discover that we have already created tables for both. You will not have to create tables for purchase orders and invoices BUT, you will probably want to add or modify some of the fields in these tables once you compare the existing Metadata against the Metadata you require for your application.

Normalized, Relational Design – Take Care

The Knowledgeone^{K1} relational database is the product of years of research and design and many, many man months of testing. It has been optimized to handle high transaction loads and large numbers of online users. It is relational and does not duplicate any field.

The real danger of making unnecessary changes is that you may affect the performance of the database and in extreme cases, threaten database integrity (by duplicating fields unintentionally).

The good news with Knowledgeone^{K1} is that the customer is empowered to modify the Data Model without programming. However, any changes still need to be designed by someone who understands relational database design and then only after taking the time to fully understand the standard Knowledgeone^{K1} database design.

Existing 'Out-of-the-box' Personalities

Before developing a whole new application (we call it a 'Personality') in Knowledgeone^{K1} first check to see if there isn't an existing Personality you can use.



For an explanation of the 'out-of-the-box' Personalities in Knowledgeone^{K1} please open the Knowledgeone^{K1} help screens and search for "multi-personalities".

A screenshot of a web-based help search interface. On the left, there is a search box with the text "multi-personalities" and a "GO" button. Below the search box is a list of search results, with "Using K1's Multi-Personalities" highlighted. On the right, there is a preview of the selected article, showing the title "Using", a sub-heading "Overview", and some introductory text. The page number "80-100" is also visible in the preview.

Type in the word(s) to search for: <input type="text" value="multi-personalities"/> <input type="button" value="GO"/>	Using
Asset Management	Overview
Changing K1 - Q&A	When v
Complaints Management	mean th
Configure by type code	Person:
Customer Relationship Management	Interfac
Electronic Document Management	
Help Desk	To illust
Human Resources Management	box'. It
Imaging	creating
Lists	
Records Management	
Tables and Personalities	
Using K1's Multi-Personalities	80-100
Welcome	We des
Where do I begin?	

Extract from K1 help screens

- [Records Management](#)
- [Electronic Document Management](#)
- [Imaging](#)
- [Workflow](#)
- [Human Resources Management](#)
- [Asset Management](#)
- [Help Desk](#)
- [Complaints Management](#)
- [Customer Relationship Management](#)

Extract from K1 help screens

There is an explanation on each of the above explaining how it works and what Triggers or Stored Procedures we have used to implement 'additional' business logic.

Please try to utilize one of the existing 'out-of-the-box' personalities before creating a new one. We 'pre-built' these Personalities to save you time and effort.

How Do You Know Which Tables To Select For Each Personality?

For example, when you configure a user as an 'Asset Manager'; how do you know which Knowledgeone^{K1} tables an Asset Manager should have access to (so you can configure the correct Security Group)?

The correct answer is that we have provided a guideline but that it will probably vary from organization to organization.



Please open the Knowledgeone^{K1} help system again and search for "Tables and Personalities" as follows:

The screenshot shows a search interface with a search bar containing 'tables and personalities' and a 'GO' button. Below the search bar is a list of search results, with 'Tables and Personalities' highlighted. To the right of the search results is a preview of the help article titled 'Tables and Personalities'. The article text includes: 'Please read [Using K1's Multi-Per](#)', '**NB:** This is a general guide only.', 'The following table indicates which', 'The list does not include link tabl', 'Those tables marked in yellow ar', and 'The K1 Administrator should use what tables each 'Group' of users'. Below the text is a table structure with a header row and one empty data row.

Extract from K1 help screens

The table in this help topic indicates which tables each Personality would usually require.

Planning Your Security Regime

Overview

If you are going to use our 'out-of-the-box' Personalities or create your own new Personalities the first thing you need to think about is security. In particular, you need to decide on Security Groups.



There is an explanation of how to Add and Modify Security Groups in the help screens. Click on the icon and then search on "The DRM Wizard" and then page through to:

Step 8: Manage Security Groups

Why Are Security Groups Necessary?

Every user in Knowledgeone^{K1} is linked to a Security Group. The Security Group determines what tables and fields (yes, security is field sensitive) you can see and what Methods (e.g., Add, Delete) you have access to when within those tables.

Additionally, your Security Group determines what Security Codes you have access to. Every object in Knowledgeone^{K1} has a Security Code, it is a mandatory field. If a record in a table has a Security Code of "Director" and your Security Group does not give you access to this Security Code, then you CANNOT see any record with this Security Code no matter that you have access to the table.

So, you can use Knowledgeone^{K1}'s security regime to control access to tables, fields, methods and even records within a table that you have access to.

Now you realize why it is important to define what tables belong to what Personality, (see section above). Once you do that, you then need to set up the Security Group for that Personality and provide access to all those tables that a Personality requires.

Why Do You Need Field Level Security?

Let's use the Human Resources (HR) Personality as an example. What if we want to add salary to the UserProfile record? But, we only want the HR Manager to see this, not the other workers in HR because it is a 'sensitive' field.

To do this we set up two Security Groups, one for the HR Manager and one for HR Workers. The big difference is that the HR Manager has access to all the fields in the UserProfile record while the HR workers only see a subset of the fields in the UserProfile record (i.e., they can't see the new salary field).

Why Do You Need Security Codes?

Let's use the example from above, our HR system. Let's assume that we only want the HR Manager to see the records of the company Directors and not the normal HR workers.

We add a security code called "Directors" and give access to this Security Code to the HR Manager but not to the HR workers.

When the HR Manager adds UserProfile records for Directors she creates them with a Security Code of "Director".

When the HR Manager looks at the UserProfile table she can see all records including those with a Security Code of Director.

When the HR workers look at the UserProfile table they only see a subset of the records in that table. That is, they cannot see any record with a Security Code of Director.

*****Getting your Security Groups right is fundamental to making Personalities work properly in Knowledgeone^{K1}.**

Changes Via The DRM

Overview

Using the DRM you can Add tables, Modify tables and Delete tables. Note that some tables are 'protected' because they are essential to Knowledgeone^{K1} operation. Generally, you can only Add, Modify and Delete non-system tables, that is, application tables.

You can also Add, Modify and Delete fields, change field sort orders (when displayed in the UI), Add and Modify column headings, change field properties, etc. This includes making a field visible or invisible.

Additionally, (but not covered in this paper because it comes under the general heading of "Changing Business Processes"), you can:

- Add, Modify or Delete database Triggers; and
- Add, Modify, and Delete Scheduled Tasks (Stored Procedures).

We make it really easy to add the tables and fields you require for your Personalities BUT, you need to think about your design and strictly control the ability of your users to add and modify tables and fields. Access to the DRM (i.e., making someone a Knowledgeone^{K1} Administrator), should only be given to people with database design experience.

Detail – Knowledgeone^{K1} Help Screens

The online help screens in Knowledgeone^{K1} contain a detailed explanation of how to use the DRM to modify the Knowledgeone^{K1} Data Model.



Please logon to Knowledgeone^{K1} or the DRM and click the help button. The DRM will take you direct to the topic titled "The DRM Wizard". Knowledgeone^{K1} will open Help at the beginning, enter "The DRM Wizard" in the search box and click OK.



Then select "The DRM Wizard" from the list at the left.

The screenshot shows a search interface on the left with a search box containing "the drm wizard" and a "GO" button. Below the search box is a list of search results, with "The DRM Wizard" highlighted. The main content area on the right displays the title "The DRM Wizard" and a "Note: You MUST have been" followed by "DRM Functions". The text continues: "The DRM is the main tool for" and "Whereas many administrative (the system tables) it is far e". Below this, it says "In v1.0 of K1 (we will extend" followed by a numbered list: "1. Backup the K1 data", "2. Manage K1 Security", "3. Manage User Profiles", and "4. Manage (add, modify".

Extract from K1 help screens

You should also study other important topics displayed on the left (see screen above) such as:

- Changing K1 – Q&A
- Making Changes to K1
- Making Changes via the DRM

Fundamentally, the DRM is actually very easy to use because we have designed a wizard and put it in front of SQL Server and Oracle and hidden the complexities. We have made the process of adding and modifying tables and fields as easy as it could be.

Tips On Using The DRM

1. We allow you to make an existing field larger but not smaller, why?

Because making it smaller would have involved making complex decisions about how to 'shorten' any existing data in this field and no matter how we did this; you would not have been 100% happy with the result.

2. We don't allow you to change the data type (e.g., from Varchar to Date) of an existing field, why?

Because data transformation from one type to another is not always possible and because even when it is possible, the results may be unpredictable and certainly not what you had envisioned.

3. After Adding a new table or new field in the DRM only the Administrator Security Group has access to it, why?

We are good but we can't read minds. We therefore chose a reasonable default (i.e., so you could see it). After Adding a new table or field you then MUST use the DRM to make it visible to the Security Groups that need it.

4. Why are some fields in tables 'Locked'?

Because they have been determined to be mandatory fields and we don't want you changing these fields because there is logic somewhere in Knowledgeone^{K1} that relies on these fields being there.

5. Why are some fields in tables 'Essential'?

Because there is logic in Knowledgeone^{K1} that uses these fields and we need to control how you Modify them.

6. Why are tables in Knowledgeone^{K1} classified as Application, System Essential or System Locked?

For the same reason we classify fields and control your ability to modify and delete them. Some tables are classified as Locked or Essential because they are essential to the operation of Knowledgeone^{K1} and we don't want you accidentally changing or deleting them. We are stopping you 'accidentally' making Knowledgeone^{K1} unusable.

7. Why must all Data Model changes be done through the DRM and not using other database tools?

See 1 to 6 above. Even though we give you the ability to modify the Data Model, we **must** manage this process otherwise you would very quickly have a crippled application.

Changes Via The UI

Overview

Using the UI you can:

1. Make FieldLinks visible or invisible
2. Create Auto Number Formats
3. Create Auto Number Format Multiple Sequences
4. Add Filters
5. Make Fields Type Code dependent
6. Make FieldLinks Type Code dependent
7. Add Forms to Knowledgeone^{K1}

Additionally (but not covered in this paper because it comes under the general heading of "User Sensitivity"), you can:

1. Change Backgrounds
2. Change Fonts
3. Add Languages
4. Change Icons
5. Change Buttons

Detail – Knowledgeone^{K1} Help Screens

The online help screens in Knowledgeone^{K1} contain a detailed explanation of how to use the UI to modify the Knowledgeone^{K1} Data Model.



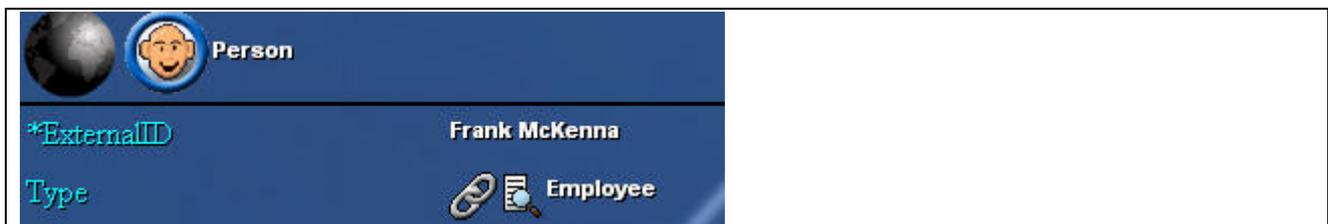
Please logon to Knowledgeone^{K1} and click the help button. Please begin with the topic "Changing K1 – Q&A". Additionally, please study the following chapters on making changes to Knowledgeone^{K1} via the UI.

Making Changes To Knowledgeone^{K1} Via The UI

Make Fieldlinks Visible Or Invisible – Seeing If A Fieldlink Already Exists

Overview

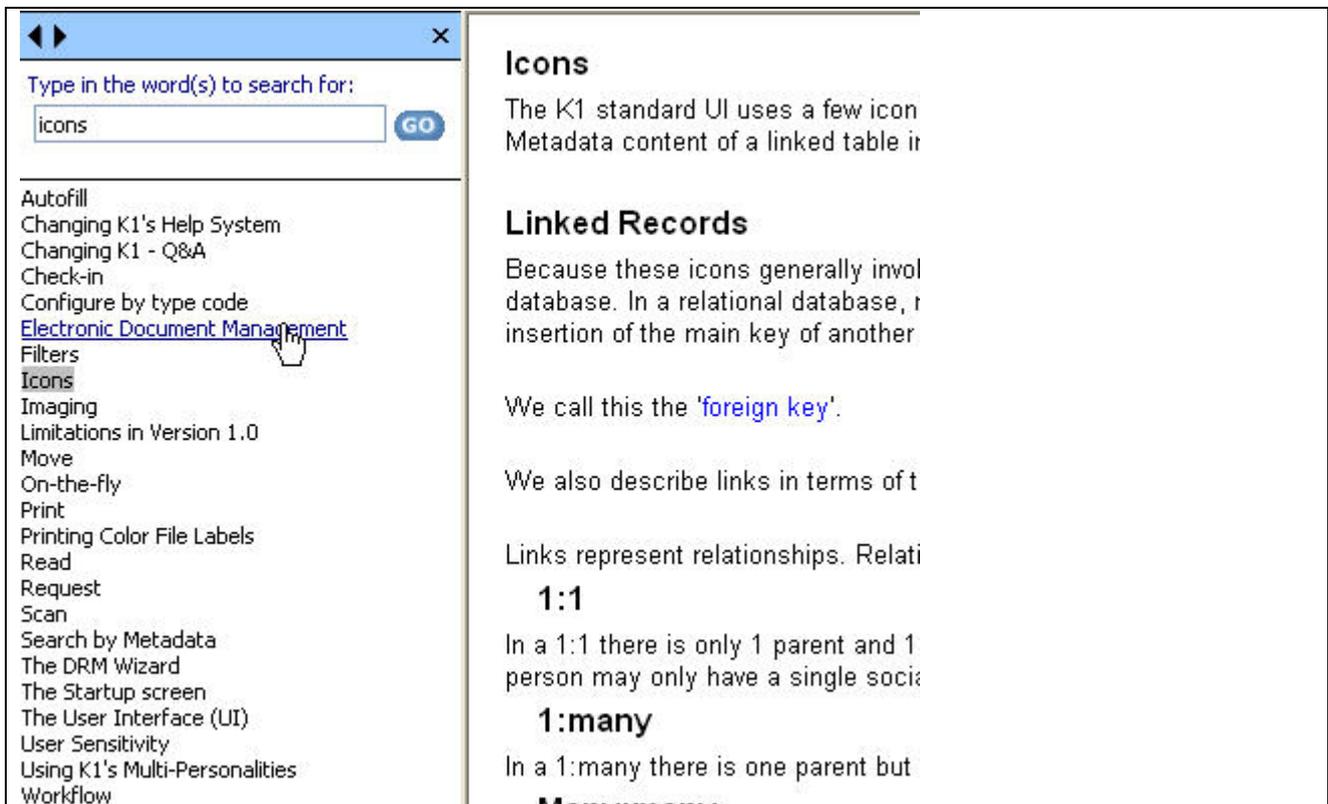
A FieldLink is how we link a field in one record to another record, usually in two different tables. For example, every table in Knowledgeone^{K1} has a FieldLink to the Type table, e.g., the following Person table record has a Link to the Type table and 'Type' "Employee" has been selected.



There are three different kinds of links between tables in Knowledgeone^{K1} and these are the only links supported.



For an explanation of the links used in Knowledgeone^{K1} please again refer to the Knowledgeone^{K1} help screens. Search on "icons" as follows and read the topic covering linked records as follows:



Extract from K1 help screens

The three types of links are:

1. Foreign key in the parent
2. Foreign key in the child
3. Foreign key in a linktable (a linktable is a 3rd table joining two tables)

A **Foreign key** is the main ID of the record you want to link to. That is, every record in the database has a unique key that the software uses to identify and access that record. In Knowledgeone^{K1}, this is a special number that is auto generated by Knowledgeone^{K1} as each new record is added. It is called the Internal ID and it is not displayed in Knowledgeone^{K1} screens but is visible when you are using the DRM.

Note: It is very important when you are adding links in Knowledgeone^{K1} that you follow the linking method already employed for existing tables and fields.

1:many

For example, when we link an Entity record to a Person record the Foreign key is in the Person record, the 'child'. This is because it is a 1:many relationship. One Entity record may be connected to many Person records BUT, one Person record may only ever be linked to a single Entity record.

1:1

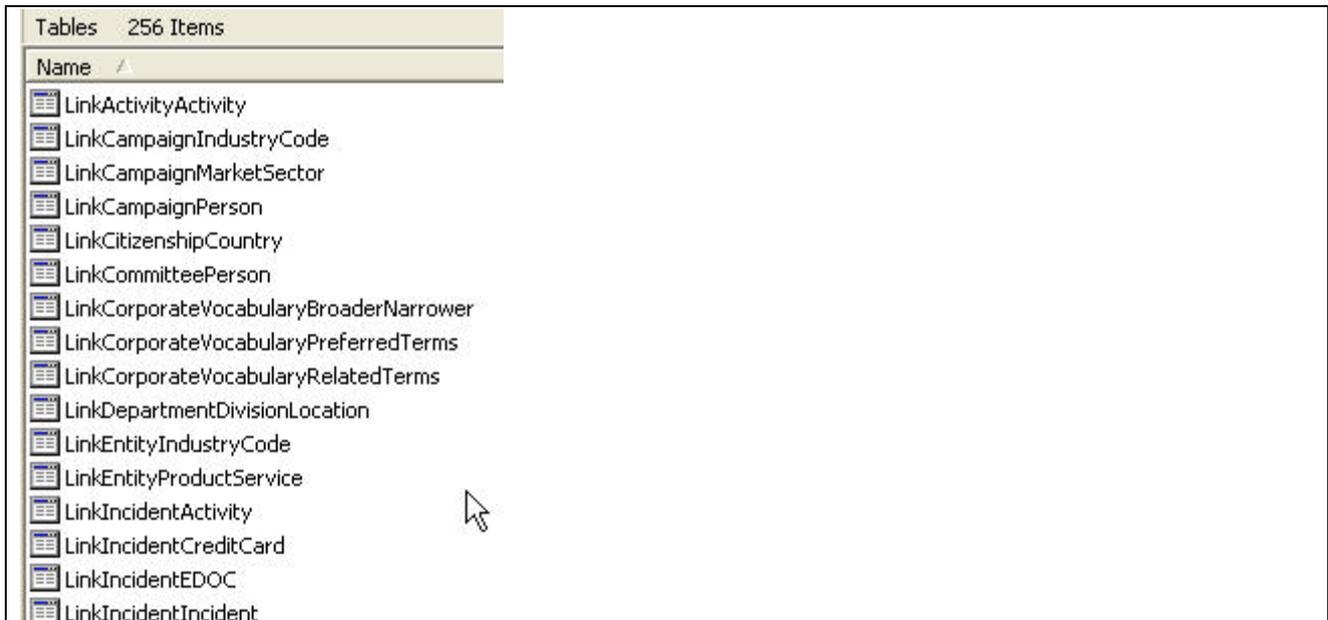
As a second example, when we link an Entity record to the Type table the foreign key is in the Entity table. This is because the relationship of an Entity to a Type is a 1:1. That is, and Entity record may only have one Type code. But, a single Type code may be linked to multiple Entity records.

Many:many

As a third example, when we link an Entity record to the IndustryCode table the foreign key is in the link table (LinkEntityIndustryCode table). This is because a single Entity record may be linked to multiple Industry codes and a single Industry Code may be linked to multiple Entities. We can call this a many:many relationship, (yes, I know, to be pedantic the relationship of the Entity record to the Industry Code records is a 1:many and the relationship of the Industry Code to Entity records is also a 1:many).

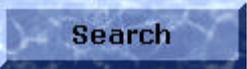
There are many link tables in Knowledgeone^{K1} (see examples below) and you are able to add and modify link tables within the DRM.

*****Are you beginning to understand why you should only allow people with database design experience to modify tables within Knowledgeone^{K1}?**



Process – Seeing If A Required Fieldlink Already Exists

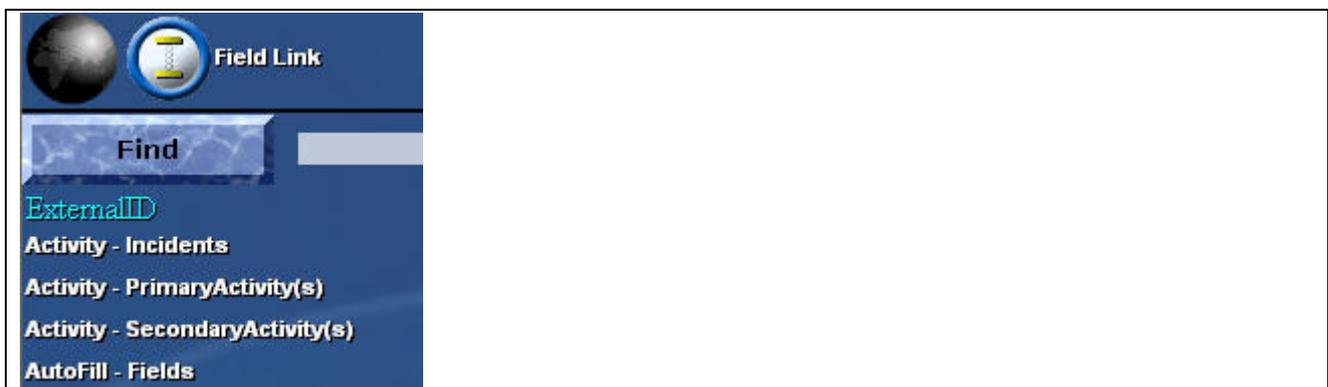
Select the FieldLink table 

Click 

Click 

You will be presented with a list of all FieldLinks.

Note that the nomenclature used is 'tablename.fieldLinkname' and that the list is in alphabetical order by table name.



****Note:** We are checking to see if a FieldLink already exists that is not 'turned on', (so we do not have to create a new FieldLink within the DRM).

Example

I am using Knowledgeone^{K1}'s Workflow and I want to add a link to the Person table that will link to all Workflow Tasks each person has been assigned as a 'Delegate'.

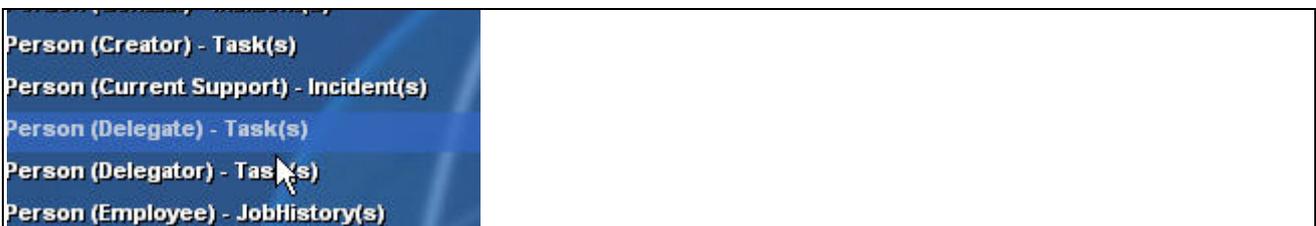
This is so when I look at an employee's Person record I can easily and quickly click on a link to see all Tasks that Person has been assigned.

The default Person record in Knowledgeone^{K1} does not appear to have this link because it is not visible when I View a Person record. Before adding a new FieldLink in the DRM I first check to see if there is one I can simple 'turn on'.

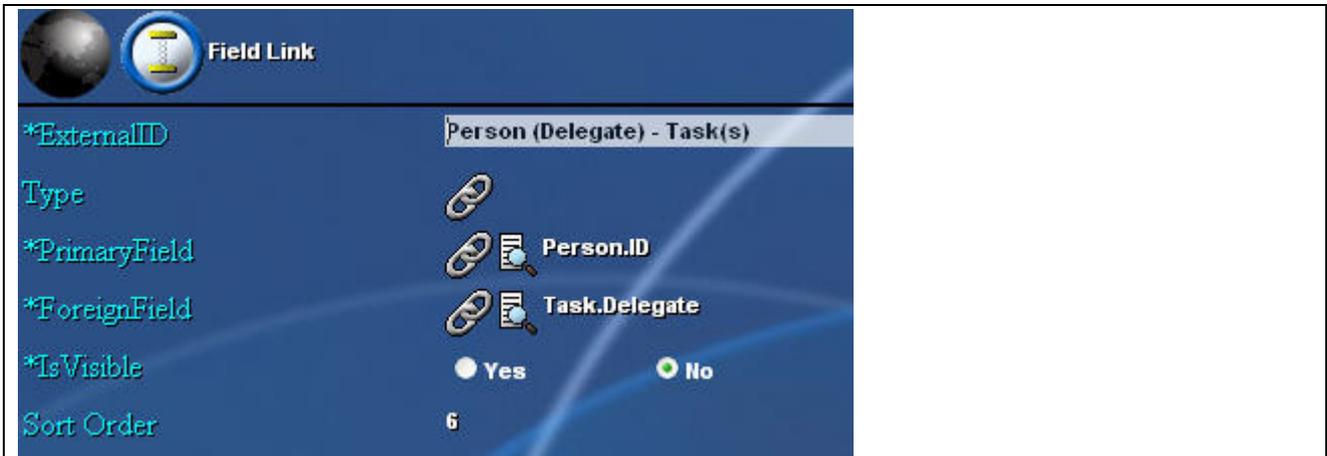
Enter "Person*" in the Find field and click the Find button.



Navigate down the list until you see the following FieldLink:



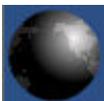
Click . You are presented with the following screen:



So, there is an existing FieldLink I can use and all I have to do is make it 'visible' for it to appear in the Person record.

Click the Yes radio button next to the "*IsVisible" field.

Click the  button.

Now click the spinning world  to return to the main screen.



Select the Person Table



Click



Click

Select or Add a Person record and View it. Notice how you now have a link to the Task table.



This link will only display Tasks for this Person.

You can also use this link to Add or Modify Tasks for this Person.

Summary

Before creating any new FieldLink in the DRM you should **ALWAYS** check to see if a FieldLink already exists.

Create Auto Number Formats

In Knowledgeone^{K1} you can allocate an Autonumber to any field. This means you can have Knowledgeone^{K1} automatically generate the next number in the sequence every time a new record is added.

Typically, you would use this feature for External IDs on things like Incidents (managing your help desk calls or complaints) or for fields containing barcode numbers.

In order to assign an AutoNumber Sequence to a field you first need to use the DRM to tell Knowledgeone^{K1} that this field will be auto-populated using an 'Autofill'. You then select the Autofill type as 14 (Autonumbered field), see the following table.

Note: An AutoNumber Format is a type of Autofill. There were 16 when the help screens were written but a 17th was added just before release. They are:

#	Description
1	Today's Date
2	Loggedon Person
3	Last Selected Object's External ID
4	Last Selected Object's Internal ID (Foreign Key)
6	The Previous Movement
10	The Table ID that the last selected Object belongs to
11	The last record from the linked table
12	A month from today's date
13	Loggedon Person's Security Code
14	Autonumbered field
15	Midnight
16	Zero Value

Extract from K1 help screens

17	True Value*
----	-------------

*Autofill value 17 is used to default a bit (Yes/No) field to Yes, because normally they start with No selected.

What Options do you have when adding an Auto Number?

- A = Letter (A - Z)
- 9 = Number (0 - 9)
- * = Any single character
- S = Sequence
- D or DD = Day
- M or MM = Month
- YY or YYYY = Year

Extract from K1 help screens

You can create any number of Autonumber formats and you can assign them to any field you wish.



And again, yes this is covered in detail in the online help screens. Search on "autofill" and then page down until you come to the sub-topic titled "Specifying an Autonumber Sequence", as follows:

The screenshot shows a search interface with a search bar containing 'autofill' and a 'GO' button. A search results list on the left includes 'Autofill', 'Changing K1 - Q&A', 'Clone', 'Configure by type code', 'Customer Relationship Management', 'Help Desk', and 'Making Changes to K1'. The main content area displays the 'Autofill' topic, featuring an 'Auto Fill' icon and introductory text: 'Autofill is where you can spec' and 'The types of AutoFills currentl'.

Extract from K1 help screens

Specifying an AutoNumber Sequence

Now let's see how we actually specify an Autofill and AutoNumber Sequence.

Task

We will specify an autonumber sequence for the Barcode number field of the MetadataProfile Type Code to be a File Folder.

Barcode#

Select the AutoFill table from K1's main screen

Select Search

Extract from K1 help screens

Create Autonumber Format Multiplesequences

Overview

A multi-sequence is when we want a single Auto Number Format to generate more than one auto-sequence number series. For example:

AB/SSSSSS
AC/SSSSSS
AD/SSSSSS

Using the above autosequences, the user enters the first two characters (e.g., AB or AC or AD) when adding a new record and then Knowledgeone^{K1} generates the next sequence number. Beginning with 1 in each case. So we begin with:

AB/000001
AC/000001
AD/000001



Please refer to the same help screens as above, e.g., Search on "autofill" and then page down until you come to the sub-topic titled "Specifying an Autonumber Sequence."

Add Filters

Overview

Filters are a way of 'filtering' the records displayed when you click on a link field. For example, if I am using the CRM Personality of Knowledgeone^{K1} and am in an Entity record (representing a company) and want to see all of the employees of this company I click the Employee link.



And this then takes me to the Person table.

No-Brainer Question

Now, when Knowledgeone^{K1} displays the records in the People table do I want to see all records (regardless of which company they work for) or do I just want to see the People records of people that work for a specific company, i.e. the company record I was in when I clicked the link icon?

- If you answered "all records" you shouldn't be working on Knowledgeone^{K1}.
- If you answered "just the people that work for that company dummy!" then you get an early mark.

Why Use Filters?

Filters are a way of ensuring that a user sees just what he or she needs to see. In the above example, there is no point in displaying 10,000 person records (the people that work for you and the 300 or so companies you deal with) when all the user really wants to see is the people that work for Knowledgeone Corporation.



Once again, Filters are covered in detail in the Knowledgeone^{K1} helps screens. Please search on "Filters" as follows:



Extract from K1 help screens

Make Fields Type Code Dependent

Overview

Let's first talk about the Knowledgeone^{K1} concept of being able to configure tables by Type Code.

Why Do We Need To Be Able To Configure Tables By Type Code?

Because Knowledgeone^{K1} is a generic application system (able to process multiple applications concurrently), we need to be able to use the Data Model as efficiently as possible. We also need to keep it as simple as possible and this means not adding unnecessary tables if we can help it.

This usually means using a single table for multiple Personalities. But, different Personalities have different requirements in terms of the fields and captions required. This is why we have provided the Configure by Type Code functionality.

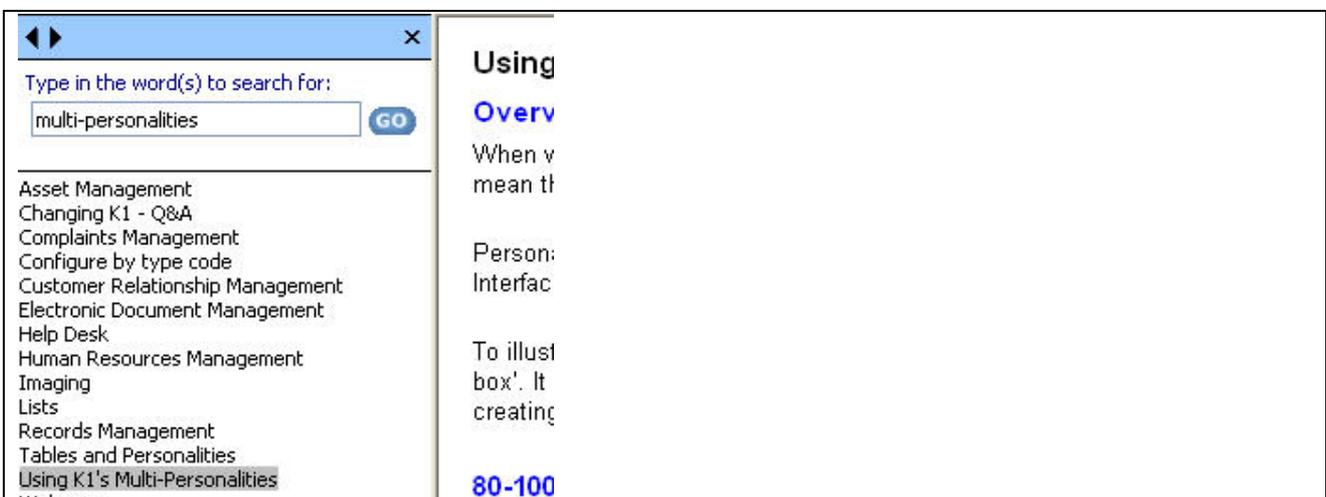
When we configure a table by Type Code we are creating different 'views' of that same table. This is so when users from different Personalities access the same table they see different things. To them, they appear to be looking at multiple tables when in fact they are looking at a single table. The best example of this is the **MetadataProfile table** or MDP table.

The MDP table has been specifically designed to hold Metadata for any object. Typically it would be used to hold the Metadata for records management and document management applications, storing all the information 'about' physical and electronic records.

In the 'out-of-the-box' Personalities of Knowledgeone^{K1} we use the MDP table for both records and assets. So, both the Records Management Personality and the Asset Management Personality are MDP table centric.



Please refer to the help system, and search on "Multi-personalities" as follows for an explanation of Knowledgeone^{K1} Personalities and their 'table-centric' nature:



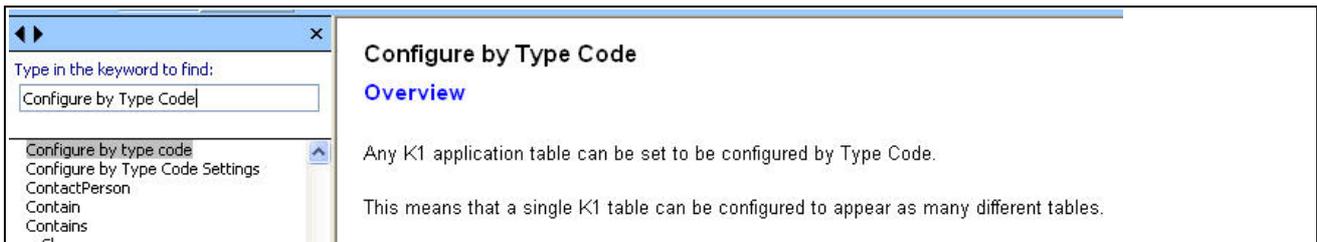
Extract from K1 help screens

When you configure a table by Type Code you are able to configure the fields that should be 'visible' to the users (e.g., hiding fields that are irrelevant to each Personality) and you are also able to change field captions, making the 'name' of a field more relevant to the users of each Personality.

And of course by using Security Codes, you can ensure that users in each Personality only see the records relevant to each Personality. That is, you use Security Codes to 'filter' the records that each user sees.



For a detailed explanation of configuring tables by Type Code please refer to the Knowledgeone^{K1} help screens. Search on "Configure by Type Code" as follows:

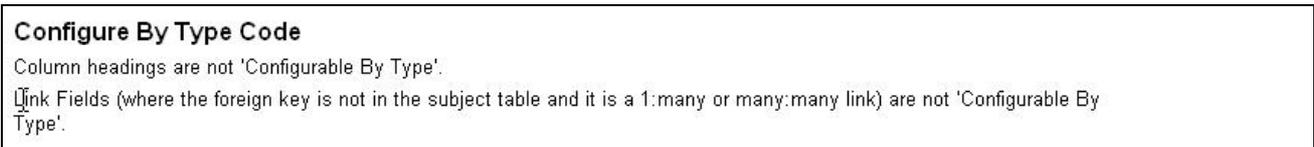


Extract from K1 help screens

Make Fieldlinks Type Code Dependent

Correction to Help Screens

Under a section titled "Limitations in 1.0" we say:



Extract from K1 help screens

This is no longer true; FieldLinks can be configured by Type Code just as Fields can be configured by Type Code. This feature was actually completed and released in version 1.0.

The process is identical to configuring Fields by type. The only difference is that you use the 'Type Dependent FieldLink Info' table rather than the 'Type Dependent Field Info' table. See below:



Why Do We Have Both Tables?

Because any record contains just two types of information. That is, information that is part of the record (a Field) and information that is part of another record and is referenced by a link (FieldLink).

In order to create a completely different view of a table we need to be able to configure both types of field.

Adding Forms to Knowledgeone^{K1}

Overview

When IT people and analysts talk about 'forms' they can be referring to many different things. This is an ongoing problem with the IT industry, the same term or acronym means different things to different people.

For example, there are some packages that integrate with workflow systems that provide electronic forms capability. There are also forms created as templates in packages like Microsoft Word that can be preconfigured to have key fields autopopulated by an application and then printed.

Knowledgeone^{K1} provides both types of forms but, as usual, in our own unique fashion; we don't use any third party forms packages.

Basically, forms are used to either enter information into a database or to receive information from a database. For the sake of clarity, let's call these two types of forms:

1. Data-entry forms; and
2. Data-output forms.

Data-Entry Forms

Surprise, surprise but almost every screen in Knowledgeone^{K1} is a data-entry form.

Let's say you are working with Knowledgeone^{K1}'s HR system Personality and you want to add a form for your employees to key in additional details.

1. Decide on the information you need, i.e., the fields.
2. Decide on the field data types, e.g., date, date&time, integer, text, etc.
3. Decide which parent table you want to link this 'form' (really a new child table) to, in this case, the UserProfile table.
4. Add your new table and new fields using the DRM.
5. Add a field in the new table specified as a link field to the 'parent' table, (the UserProfile table).
6. Use the DRM to change the Security Group so the employees now have access to this new table and just the fields you want them to enter.
7. Either email each employee with instructions or, setup a Workflow in Knowledgeone^{K1} and a single Task. Clone this Task changing just the Delegate field in each case and have Knowledgeone^{K1} automatically email the employees telling them about the Task to be completed. If you choose the latter approach you can then also use Knowledgeone^{K1} to monitor and manage the exercise, telling you instantly who has completed the new form and who hasn't.



For more information on using Knowledgeone^{K1}'s Workflow functions please refer to the help system. Search on "Workflow" as follows:

The screenshot shows a search interface with a top navigation bar containing 'Contents', 'Index', and 'Search'. A search box on the left contains the text 'workflow' and a 'GO' button. Below the search box is a list of search results, with 'Workflow' highlighted. The main content area on the right is titled 'Workflow' and contains the text: 'Please read [Using K1's Multi-Personalities](#) before reading this topic.' Below this text are two buttons: 'Work Flow' (with a train icon) and 'Task' (with a hand icon). At the bottom of the main content area, there is an 'Overview' section with the text: 'Workflow in K1 is Workflow table centric.'

Extract from K1 help screens

Data-Output Forms

Surprise, surprise, you can create any form you need using Knowledgeone^{K1}'s Print Method.

1. The Print function is based on the Active Reports engine. We purchased a developer copy and embedded this in every copy of Knowledgeone^{K1}.
2. You use the Active Reports Designer (exposed within the Print Method) to design a form or any number of forms to your unique requirements. You save each form in Knowledgeone^{K1} in the Saved Report table (so it can be re-used).
3. You select the record set (i.e., the selection of records that you wish to use to populate the form) using Knowledgeone^{K1}'s standard Search Method, select to use a Saved Report (e.g., Annual Leave forms), and then simply produce the forms with all of the required fields automatically populated by Knowledgeone^{K1} from the record set produced by the Search.

Note: If you are not an experienced Active Reports user you will need to buy a single copy of Active Reports (to get the full manual) so you can better understand how to utilize the embedded Active Reports engine to produce the forms you need.



For a detailed description of the Knowledgeone^{K1} Print Method please refer to the help system. Search on "Print Method" as follows:

The screenshot shows a search interface with a top navigation bar containing 'Contents', 'Index', and 'Search'. A search box on the left contains the text 'print' and a 'GO' button. Below the search box is a list of search results, with 'Print' highlighted. The main content area on the right is titled 'Print' and contains a 'Print' button (with a blue gradient and a mouse cursor over it). Below the button is the text: 'Unlike conventional applications, there are no 'standard' reports in K1. Instead K1 comes with a Report Wizard that allows you to quickly and You can also save the report formats you create so they can be used :

Extract from K1 help screens

Active Reports

The report engine in K1 is a third party product.

We used Active Reports for .NET Professional Edition V2.0 1 Developer License (for Visual Studio .NET 2003) in v1.0 of K1.

Active Reports permits a royalty free distribution of its report engine when embedded in an application like K1 (because K1Corp bought a developer license).

Should you wish to have your own copy of Active Reports (and enjoy more functionality than is available via the report wizard in K1) you can contact them at:

www.componentsource.com

Tables and Methods

“I have added a new table but now I can’t Modify it. Why?”

You have to remember that the Knowledgeone^{K1} Administrator determines what Methods are applicable to each table. This is done in the DRM.

The Knowledgeone^{K1} Administrator can also modify this master setting for each Security Group, again within the DRM.

So, if you have created a new table but don’t seem to be able to Modify it, it will be because you have either not yet set the Methods for this table in the DRM or, your Security Group does not include the Modify Method for this new table.



Please refer to the Knowledgeone^{K1} help screens. Search on the “DRM Wizard” as follows:

The screenshot shows a web-based search interface. At the top, there are navigation tabs for 'Contents', 'Index', and 'Search'. Below the search bar, the text 'Type in the word(s) to search for:' is followed by an input field containing 'the drm wizard' and a 'GO' button. A list of search results is displayed on the left, with 'The DRM Wizard' highlighted. The main content area on the right displays the title 'The DRM Wizard' and a note: 'Note: You MUST have been...'. Below this, the section 'DRM Functions' is introduced, followed by a list of five numbered steps: 1. Backup the K1 data, 2. Manage K1 Security, 3. Manage User Profiles, 4. Manage (add, modify) Tables, and 5. Manage Database Tables.

then page down to:

Step 10: Manage Tables and Fields

and

Step 8: Manage Security Groups

for a detailed explanation of how to configure Methods to Tables.

Data Dictionary

“How do I find out what tables and fields are in Knowledgeone^{K1}?”

Because the customer can modify Knowledgeone^{K1}'s Data Model there is no static 'Data Dictionary' as there would be in a conventional application (i.e., with a fixed Data Model). However, because K1 is an 'open' application, exposing all of its details in its tables, it is very easy to inspect the Knowledgeone^{K1} Data Model.

Note: Unlike conventional applications, Knowledgeone^{K1} uses Knowledgeone^{K1} to manage Knowledgeone^{K1}. What this means is that Knowledgeone^{K1} exposes most of its 'workings' within its system tables. In a conventional application, these 'workings' would be hidden in compiled code.

So, Knowledgeone^{K1} has a table called the **Table** table and a table called the **Field** table and a table called the **FieldLink** table.

As you modify Knowledgeone^{K1} using the DRM, these three tables are automatically updated so as to contain an accurate model of all tables and all fields within Knowledgeone^{K1}.

****** This is another reason ALL Knowledgeone^{K1} data Model changes MUST be done via the DRM. The DRM will automatically update the Table table, the Field table and the FieldLink table every time you make a change.**