

Entities

- Interfaces
 - Understanding interfaces
 - Using interfaces
- Questionnaires
 - Step-by-step questionnaire for anonymous users
 - Step-by-step questionnaire for specific users
 - Controlling the status flow
 - Questionnaire building process
 - Setting up a questionnaire with report
 - Cheatsheet
- Status
 - Status Actions
 - Status Notifications
- Entity Settings

Interfaces

Understanding interfaces

Interfaces allow access to the system with preset values for

- Wrapper
- Stylesheet
- Template
- Language
- User context

When accessing the interface the user is redirected to the main page with the correct parameters set. In case the user context is active, a session with a matching user profile is also created in the process.

The following types of record operations are allowed

- Create new records
- Update an existing record

Create operations

Create operations require no additional security information and can be used directly.

URL for testing the interface:

`http://[server name]/[application]/webinterface?[interfacename]`

Update operations

When updating an existing record a security token must be passed along to the interface:

- The right data is displayed for the user
- Bind user session to this piece of data

A security token contains

- a reference to the record
- a timestamp when the token was issued
- a signature that guarantees that the server issued the token

URL for testing the interface: `http://[server name]/[application]/webinterface?[interfacename]=[security token]`

Security tokens can be generated in various ways

- Create a status action that generates a notification
- Administrator tool: Designer > Modules > Admin Services > BuildSecurityToken
- Add the special "Token display" field to a solution

Questionnaires and email invitations

Generation of security tokens are closely tied to the use of questionnaires and status actions.

The guide [Questionnaires](#) covers such an example.

Using interfaces

Interfaces are access points for external users (someone using the system without a login).

For a more detailed description, please read: [Understanding interfaces](#)

Basic configuration

An interface has the following configuration options

- Interface URL
 - Unique name of the service
- How should the interface behave
 - Look and feel:
 - Wrapper: "Outer" page design, possibly mimicing an existing website
 - Template: "Inner" page design
 - Stylesheet: Color, font, styles etc.
 - Language: Default language set on page
 - Use questionnaire: Use multiple pages in stead of one big page
- What is the external user allowed to do
 - Proxy user profile
 - Allow UPDATE of existing records
 - Allow INSERT of new records

Note that using an interface is no different than being a normal user: The external user will mimic the provided system user, and this user must have the appropriate permissions.

Email tokens for access to existing records

A common usage is sending email tokens to external users.

Tokens will allow a certain user to update the content of a certain record. Usually these are tied together, where the email of the user is a field in the record.

Tokens contain a timestamp, so it is possible to invalidate the token after a specified amount of time

- **Token expiry**: Maximum allowed age of the token

Please note that this use requires some additional configuration of the Status action that sends the emails.

Other interface options

Control what happens after an update or insert operation

- **Success message:** Displayed message (if not using questionnaire interface)
- **Failure message:** Displayed message (if not using questionnaire interface)
- **Redirect:** URL to send user to after updating/inserting their data

Extra security features

- **CAPTCHA:** User must take human test before using the interface (prevents robots from injecting data)
- **IP filter:** Specify which IP's are allowed to use the interface (prevent users from known problem countries)

Questionnaires

Step-by-step questionnaire for anonymous users

1. Create a new solution and test it
2. Create an interface
 - [solution] > Advanced > Interfaces ... ADD
 - Interface
 - Active: X
 - Unique name: [unique name across all interfaces]
 - Anonoumous user
 - Active: X
 - **Insert items: X**
 - Use questionnaire style interface: X
3. Set the order of fields
 - [solution] > Wizards > Questionnaire order
 - Enter values
 - 0 = disabled in questionnaire
 - Numbers dont need to be consecutive
 - Accept with Submit
4. Ensure that anonoumous users can access to the solution
 - UserID = 10003 must have a group that has permissions
5. Test link using
 - [http://\[server name\]/\[application\]/webinterface?\[interfacename\]=1](http://[server name]/[application]/webinterface?[interfacename]=1)

Step-by-step questionnaire for specific users

1. Create a new solution and test it
 - Make sure that solution contains an field of the Email type
2. Create an interface
 - [solution] > Advanced > Interfaces ... ADD
 - Interface
 - Active: X
 - Unique name: [unique name across all interfaces]
 - Anonoumous user
 - Active: X
 - **Update items: X**
 - Use questionnaire style interface: X
3. Create an status action that sends emails with tokens
 - Required: Set the correct interface
 - Option: Set a status that is used if email sending fails
4. Set the order of fields
 - [solution] > Wizards > Questionnaire order
 - Enter values
 - 0 = disabled in questionnaire
 - Numbers dont need to be consecutive
5. Ensure that anonoumous users can access to the solution
 - UserID = 10003 must have a group that has permissions
6. Test solution
 - Goto frontend
 - Create new record and set status to that of the action trigger
 - Test token
 - Collect from you own email OR
 - Check the email queue in the designer

Controlling the status flow

Status after **first update**:

1. Designer > [Solution] > Advanced > [Interface]
2. Update the value "Status after update"

Status after questionnaire **completion**:

1. Designer > [Solution] > Advanced > Functionality - Questionnaire - Messages
2. Update the value "Final status"

Questionnaire building process

In order to create a complex questionnaire you first choose between copying a solution and creating from scratch.

We strongly recommend the copying if you are unfamiliar with designing solutions.

Creating from scratch

Steps to create a questionnaire includes

1. Create solution and add fields
2. Configure questionnaire order and messages
3. Setup interface for external users
4. Configure email invitation
5. Build report template

For detailed information read the section **Setting up a questionnaire with report** below.

Copying a solution

Steps include

1. Backup existing solution
 - *Entity* > Advanced > Backup > Create new backup
2. Promote backup to standard
 - Click top backup > Promote to standard
3. Create instance from standard
 - Quickstart > Repository > *choose* > Create this item
4. Deploy solution
 - *New entity* > Deploy live

Setting up a questionnaire with report

Field types

Complex questionnaires contain fields for sending the questionnaire and answering questions.

Depending on how the questionnaire is distributed to the users, you will also need contact information such as name and email.

Questions and scores

For normal questions that are answered once use the field type: **Complex question**

A complex question contains a single question and multiple answer. Each answer contains

- Value shown when selecting
- Response showed in report
- Color and icons
- Score

You can optionally calculate scores across multiple questions using the field: **Calc: Complex question (sum)**

Support for multiple questions

Multi line questions consists of the following

- Multiline: Location editor
- Multiline: Complex question

There should only be exactly one **Multiline: Location editor**, while there can be multiple **Multiline: Complex question**.

First the users enters lines values in the location editor. Then questions are created for each **Multiline: Complex question** - one for each line entered in the location editor.

Lines can be removed by clicking the checkbox next to them in the **Multiline: Location editor**.

Questionnaire

Normal users will see data represented as a form, while external users will most often have displayed a step-by-step questionnaire.

Inclusion of fields and their page order is managed in the questionnaire order editor: **Entity > Advanced > Questionnaire: Question order**

Note that

- If a field have a sortorder of 0 (zero), it will never be displayed for external users.
- Order numbers are not displayed to the user and do not need to be consecutive - ex. 10, 20, 30 ...

The initial and final messages of the questionnaire is edited in: **Advanced > Questionnaire: Messages**

From a normal view normal users can see data in questionnaire form, by appending an extra command to the address bar in the browser: **&StepNumber=0**

Granting access for external users

The process consists of 2 activities

1. Add access for external users by adding a new permission
2. Add an interface for the solution: Advanced > Interface: Add
 - Enter a descriptive and unique name
 - Check "UPDATE"
 - Check "Use questionnaire style interface"

Sending emails with links

In order to send personalized message your solution will need to contain these fields

- Text: Enter the name of recipient
- Email: Email of the recipient

These fields are referred to in the message template in a status action, which is where subject and body of the email is edited.

A message is sent to the recipient, when a record in the system reaches a certain state.

Configuring reports

Report are Word documents containing special tags where content from the records are merged into the document.

1. Create a new Word document
2. Write content and add special field tags
 - Entity > Advanced > Template: Add > Office > (copy special tags)
3. Add template to the solution
 - Entity > Advanced > Template: Add
 - Choose Word or PDF add
 - Upload the Word file

Note that certain configuration options for fontsize, margin etc. can be configured per solution:
Entity > Advanced > Configurations

- GlyphPageWidthPercentage
- HeatmapColumnMaxCharacters
- renderExtraLinjeBeforeNotes
- renderExtraLinjeBeforePicture
- renderQuestionInTable

Often you will want at heatmap of the answered questions in first pages of the report. For this you add a field of the type: **Word heatmap index**

In case you want a dedicated button at the top of the page please add a field of the type: **Action button: Export template**

Cheatsheet

Fields in solution

Normal questions

- "Complex question"

Sum of scores in normal questions

- "Calc: Complex question (sum)"

Setting up multiple response for a question

- "Multiline: Location editor"
- "Multiline: Complex question"

Contact information for sending emails

- Text and Email for contact information

Content for the Word report

- "Word heatmap index"

Questionnaire field order

Entity > Advanced > Questionnaire: Question order

Order 0 means field is NEVER displayed

Questionnaire start and end messages

Entity > Advanced > Questionnaire: Messages

Send invitation to customer

1. Entity status: Click status > Add status action
2. Set email recipient = Email field
3. Integrate name in message: Dear {NAME}

Template for report

1. Create Word document
2. Add content tags - ex. "\${NAME}"
3. Add to solution: Entity > Advanced > [Template list] > Add

Status

Status

Status Actions

The actions are fired when a certain condition is met AND the item is in the parent status

- An item enters the status
- An item leaves the status
- Time has passed while an item was in this status

Timed actions

The time specified is relative to a value of the item

- Creation date
- Last change
- Last status update
- Dynamic **date field**

Time is specified in days and can assume both negative and positive values.

Action types

Notification

This action sends an email for users or groups.

Various options for the target email exist, including

- Raw email: Static reference
- User: A specific user in the database
- Field value: The contents of another field
 - Email
 - Phone (sent by SMS)
 - CPR nr (sent by eBoks)

Different types of content can be included in the emails

- Data from the record

- Document generated using data of the record
- Links to the record (internal users)
- Interface tokens (external users)

Read about special tags and formatting her [Tutorial/Status_notifications](#)

Task create

This will dynamically create new tasks for any '[Task list](#)' fields found in the solution.

Codeunit

This action triggers the execution of a special codeunit: [Codeunit/Statusaction](#).

Except for timed actions you would implement this inside a [Codeunit/Formevents](#).

Export

The template is optional: If no template is found raw XML will be generated (FTP/mailto will not work).

The following types of routing are supported for template based exports

- Local file system (optionally a mapped share)
- Remote FTP server
- Sent by email

Please note that export actions are handled in separate threads (performance), and there is no guarantee the operation succeeds (tjeck the event log)

Values from record

The target reference may contain field references in the {FIELD} format, that will be populated at runtime.

Other special tags include

- {SagID}
- {DataID}
- {Resume}
- {NanoTime}

Note that values in records are not filtered for illegal og troublesome characters (such as \ or .)

Usage examples

Routing	Syntax	Syntax example
File system	<i>local filesystem path</i>	c:\exportFolder\{GROUP}\{TITLE}.doc x
FTP server	<i>ftp location and connection string</i>	<u>ftp://username:password@acme.com/</u> <u>exportFolder</u>
Send by mail	"mailto:"+[email]+":"+[subject]+":"+[filename]	<u>mailto:boss@acme.com:Attention:you</u> <u>r_2013_report</u>

Shift status

A status merely changes the status to a new value.

This type of action only makes sense to use in timed actions.

Webhook

This action performs an HTTP-request with parameters from the updated item and update the item based on the result.

Datapoint		Type	Notes
url	Required	String	The url that is to be called. This supports input of parameters from the item.
method	Optional	String	The HTTP-method of the call, one of: "POST", "GET", "PUT" or "DELETE". Default: "GET".

type	Optional	String	The datatype of data returned, supported types: "json", "raw" or "XML". Default: "json".
headers	Optional	Map	A map of extra headers that should be set
params	Optional	Map	Path parameters that should be set and send.
body	Optional	String	A string that will be set and send. Not available for "GET".
update	Optional	Map	Fields that should be updated based on the data returned. If type is "raw", only the first item in this list will be updated, and it will be set to the entire response. If type is "json", attributes with keys formatted as <code>[FIELDNAME]</code> will be updated from the given key in the returned data.

In `params` and `body`, values formatted as `[FIELDNAME]`, will be filled with data from the record.

Sample

```
{
  "url": "http://localhost:3001/users/1",
  "method": "PUT",
  "type": "json",
  "headers": {
    "authorization": "Bearer XYZ",
    "Content-type": "application/json"
  },
  "params": {
    "param": "DATA"
  },
  "body": {
    "id": 1,
    "name": "[NAME]",
    "email": "john.doe@example.com"
  },
  "update": {
    "[NAME]": "name"
  }
}
```

Status Notifications

Special tags for notification

Data in emails

Dynamic field content can be inserted using `{FIELD_SYSTEM_NAME}` syntax.

For status notification content the following tags will also be populated

- `{USER}`: Name of the user
- `{RESUME}` Name of the record
- `{LINK}`: Clickable link to the record (se below)
- `{NOLINK}`: Do not include any record links
- `{RAWLINK}`: Link without any HTML markup

Note that a link to record will be included at the bottom of the message, unless a `LINK`, `RAWLINK` or a `NOLINK` placeholder can be found.

Create attachments

Additionally rendered files can be attached using the following syntax

```
[REPORT:123456]
```

The number refers to the ID of the template to render.

This will render the the template, and attach the resulting file as an attachment to the email.

Default name of the file is "attachment". In case you want a custom name add this in the tag (ex. "Foobar")

```
[REPORT:123456-Foobar]
```

Tip: It is perfectly OK to use templates with any tags in it, but in that case you might consider using the link approach below.

Files from the record can be attached using the following tag, where 123456 is the ID of the field.

```
[FILES:123456]
```

Dynamic links

URLs can be entered by using {APPURL} in front of a relative reference. Example:

```
{APPURL}/login
```

Static file links

If you only need static files, just upload them to the media library and put a link in the mail

1. Upload to media library: Ressources > Media files > Add
2. After upload right-click "show" button and select "copy link"
3. Paste the link into a notification (no html encoding is needed)

Note that this will only make a link to the file (the file is not added as an attachment).

Entity Settings

Below is the full list of settings found on any given Entity. This is the expanded view, which is available when toggling the slider in the top right. Below the image are explanations of every setting.



General

Active



System name

crmaccount [dfe13c7d8d07e0464aab8ff653ef8330]

Display name

(default)

(list views)

Description

Icon/image

Workflow

Initial status



Deleted status

(hidden)

 Recursive delete/hide (parent trigger) Satellite data synchronization

Functionality

Options

 Check for Duplicates using unique key(s) Measure time spent editing / viewing each item Allow Internationalization Offline enabled

Logging

 Use Revision log to build log of all changes Use Access log to register user access to each item Use Status log to register time in each status Use Activity monitoring (historical data) All users can display logs (Analyst role not required)

Targetting

 Monitor and analyse waiting time

Behaviour

Form options

 Display Page selector using  Display Field selector (toggle secondary fields) Display Template selector on item views Show List after update (alternative: options)Item command (default: edit)

List options

 Display Create link on list views

Top buttons

deploy live: Submits all new entity and field changes, so they are rendered in the application.

backup: Access and create new backups

delete: Removes the entity entirely. Record data is not lost from this action.

bulk update: ???

General


Active: Determines whether or not records from the entity is available in the application. It is often useful to deactivate unused entities instead of deleting them, in case you need to look at what once was.

System name: What the entity is called in the database and is sometimes used in configurations where a specific entity needs to be pointed out. An entity's system name never changes.

Display name: What the entity is called from the point of view of the user, and can be changed at any point if needed. Below is what the plural name of the entity should be for the purposes of list views.

Description: Has no function. Is merely a reminder in the designer of what the entity is.

Dev notes: A place for the developer to note things about the entity. Only used for this, is never outputted in the frontend.

Icon/image: Choose an icon to display next to the entity selection in the application navigation menu. The button **Icon-selector** below helps with this. Example:  Accounts

edit instructions, header and script: Advanced menu to insert custom css and JavaScript to adjust the appearance and behaviour of the records.

Workflow

Initial status: The status that all records should start in (before first save).

Deleted status: The status that records should enter, when the delete button is pressed. When set, the delete button will appear in the front end, and the delete method in the API will work. A status has to be marked as "Hidden data" to appear in this list.

Recursive delete/hide (parent trigger): Also delete records in this entity when a record in a parent entity is deleted.

Satellite data synchronization: Enable synchronization of records to another TS application.

Functionality

Check for Duplicates using unique key(s): This feature prevents any record from being created if another records shares its combination of unique field values. Will also prevent one record from being edited to be the same as another. This only works if at least one field is required to be unique.

Measure time spent editing / viewing each item: Logs approximate number of seconds, that a user had a record open, and what status that record was in. Required a re-deploy of the entity.

Allow Internationalization: Enables the internationalization system for this entity. Without this translations will not work.

Offline enabled: Enable offline functionallity for this entity. Eg. allows for creation of record without an active connection to the server, if the entity has been viewed once.

Use Revision log to build log of all changes: Creates a log of all future changes to the records. Includes when and by whom the changes were made.

Use Access log to register user access to each item: Creates a log of when a user opens a record in this entity.

Use Status log to register time in each status: Creates a log of every time a record changed status, including: from-status, to-status, assigned at, assigned-by and waittime.

Use Activity monitoring (historical data): ???

All users can display logs (Analyst role not required): ???

Monitor and analyse waiting time: ???

Behaviour

Display Page selector using: Selects what type of grouping the fields should be shown with.

- DEFAULT: Adds pages as tabs at the top of the record
- Tabs table (old):
- Tabs jquery (new):
- Section headers: Adds pages as collapsable sections
- Column squares:

Display Field selector (toggle secondary fields): ???

Display Template selector on item views: ???

Show List after update (alternative: options): Determines how many submit buttons are displayed at the bottom of the record page. If this is checked, there will be one that just saves changes and exits the page. Otherwise there will be two extra buttons that respectively show the

record in show mode and edit mode after saving.

Item command: Overwrite the "item" (default single record command) command. Default is "edit", other option is "view".

Display Create link on list views: Enable the "Create record" button on list views.

Allow Quicksearch on list views: Enables the resume quick search on lists.

Display link for Archived item search: Enables link to view "hidden" records on lists.

Display Subtable lists on list views (file, task, signature lists): Enable tabular data (not list of children) in list view.

Reverse sorting (newer/lower on top): Set sorting to ascending.

Many command: Overwrite the "many" (alternative multiple records command) command. Default is list, other option is a any multi-record view, eg. calm.

Hide solution in Main menu: Hides record list link from the application navigation menu.

Solution section: Choose which section the entity should belong to in the application navigation menu.

Sort order: Defines sort order in the menu for the section, that this entity is in.

Integration

Codeunit: Advanced feature to add a reference to a TS codeunit to add complex behaviour.

Questionnaire: ???

Generate default OLAP schema: ???

Generate standard permissions for default OLAP schema: ???

??? : is granted full access to default OLAP schema

Allow REST webservice interaction: Must be enabled to access records via rest. A Specific Policy must be enabled for this to work.

Models

Model is active: ???

XLSX file: ???

Reader is active: Enables AI processing of records. This includes bulk uploading and later processing via the service runner.

Security

Use Exclusive group for access control: If this entity has a "lookup: exclusive group" field, only users in that exclusive group can see records from that group. Any users with no exclusive group are not limited this way.

Use Lists of users for each item: ???

Use Lists of groups for each item: ???

Use Creator only restriction (ignore group recommended): Only the user that created the record can access it.

Inheritance: ???

Ignore: Users in this group ignores all advanced access rules.

Share group: Only used when "Use Exclusive group for access control" is enabled. Controls whether records without an exclusive group are accessible by all users, or only users without exclusive group. And controls whether those records are locked or possible to edit.

Default

Language: Choose between established languages for records in this entity.

Style: Overwrite what set of icons this entity should use.

Stylesheet: Overwrite what stylesheet this entity should use.

Template: Overwrite what template should be used to render the record.