Agile
extensions

# Give your release notes the love they deserve

Bravo Notes is a tool for collecting, organizing, editing and publishing information about the software you are releasing.

The rise of Agile and Devops and allows leads the industry to shipping more value to customers ever more quickly which also means we have to communicate about changes in our software more frequently.

## 1 The status quo of release notes creation

So how are software companies build release notes today?

We have gotton numerous replies from you on this and at the moment there are 2 major scenarios we are seeing.

**1. Release notes are generated complete automatically via a release or build usually for internal use (as they are fairly technical)**

**2. Release notes come out an often painful process of exporting work items, some Excel munching, copy&pasting into Word for content editing, format breaking and re-fixing (and to generate a PDF in the end).**

This is overgeneralizing and your mileage may vary but you get the idea.

As we see the most pain to eliminate in the 2nd scenario we are focussing on this one for now. (We will probably offer a build task for more automation in a future version). d

## 2 Content is king

The bulk of the information we are talking about here usually lives inside work items like Features or fixed Bugs.

Some of us may just use the work items title. Others use custom fields to store release notes information on each item.

But lets be honest: Even though the gist of the information is there in the work item data almost every piece of content has to (or should) be looked over by a marketing person, business analyst, product owner, technical writer or someone else who can make sure that the final release notes have a consistent voice

Also you usually want to write some introductory notes (just like what are you reading right now) before you dive into details of new features.

## 3 Introducing Bravo Notes

Bravo Notes is still in private beta but this way we want to share with you what features we have already build and we hope to hear from you to let us know if we are missing critical capabilities and features.

These release notes were written, compiled and published 100% via Bravo Notes!

So let's get to the show and tell part!

# 4 Initial features

These features are very likely to ship with the first official release we might add, remove or change features before that.

## 4.1 Release notes templates with live preview

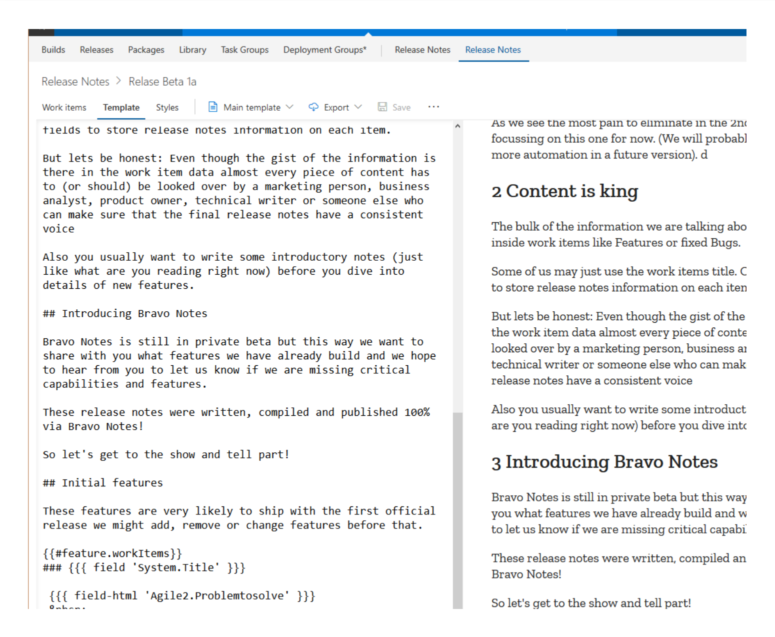
### The problem

Usually the structure and layout of your release notes dont change from release to relase that often. So ideally you want to focus on the content of the release and don't have to think about layout, formatting.

### Our solution

Bravo notes offers simple yet flexible templates designed specifically around the use cases around release notes creation.

Here is a screenshot of the template editing and preview screen.



```
fields to store release notes information on each item.

But lets be honest: Even though the gist of the information is
there in the work item data almost every piece of content has
to (or should) be looked over by a marketing person, business
analyst, product owner, technical writer or someone else who
can make sure that the final release notes have a consistent
voice

Also you usually want to write some introductory notes (just
like what are you reading right now) before you dive into
details of new features.

## Introducing Bravo Notes

Bravo Notes is still in private beta but this way we want to
share with you what features we have already build and we hope
to hear from you to let us know if we are missing critical
capabilities and features.

These release notes were written, compiled and published 100%
via Bravo Notes!

So let's get to the show and tell part!

## Initial features

These features are very likely to ship with the first official
release we might add, remove or change features before that.

{{#feature.workItems}}
### {{{ field 'System.Title' }}}

 {{{ field-html 'Agile2.Problemtosolve' }}}
```

As we see the most pain to eliminate in the 2nd
focussing on this one for now. (We will probabl
more automation in a future version). d

2 Content is king

The bulk of the information we are talking abo
inside work items like Features or fixed Bugs.

Some of us may just use the work items title. C
to store release notes information on each iten

But lets be honest: Even though the gist of the
the work item data almost every piece of conte
looked over by a marketing person, business ai
technical writer or someone else who can mak
release notes have a consistent voice

Also you usually want to write some introduct
are you reading right now) before you dive into

3 Introducing Bravo Notes

Bravo Notes is still in private beta but this way
you what features we have already build and w
to let us know if we are missing critical capabi

These release notes were written, compiled an
Bravo Notes!

So let's get to the show and tell part!

For the technical folks it might be interesting that the main template is based on

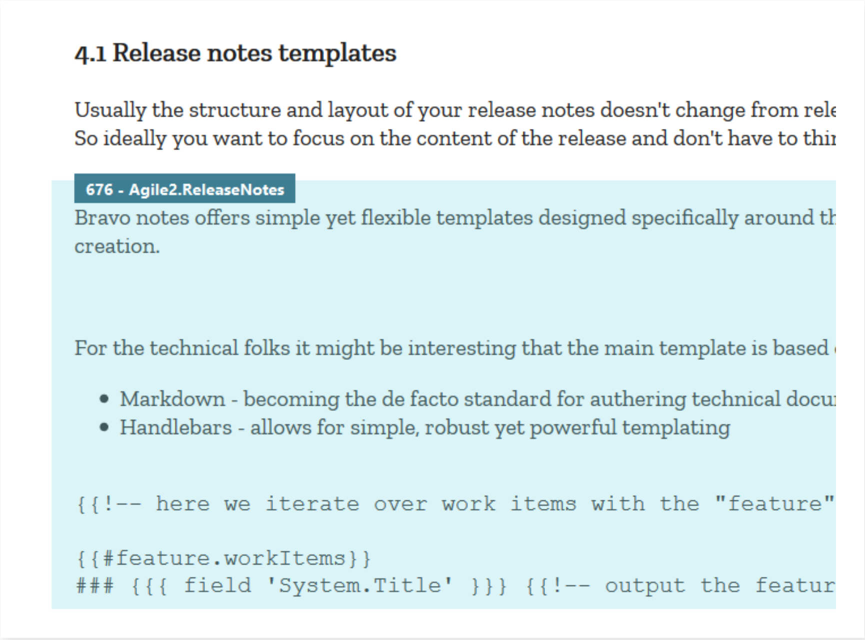
- Markdown - becoming the de facto standard for authering technical documentation
- Handlebars - allows for simple, robust yet powerful templatin

```
{{!-- here we iterate over work items with the "feature" label --}}
```

```
{{#feature.workItems}}
### {{{ field 'System.Title' }}} {{!-- output the feature title as a h3 }}

 {{{ field-html 'Agile2.Problemtosolve' }}} {{!-- and two custom... }}

 {{{ field-html 'Agile2.ReleaseNotes' }}} {{!-- ...html fields }}

{{/feature.workItems}}
```

My head just exploded due to this template inception (showing the source of the template generating a document in the document).

Anyways. The result will be rendered as shown below:



### 4.1 Release notes templates

Usually the structure and layout of your release notes doesn't change from rele
So ideally you want to focus on the content of the release and don't have to thir

676 - Agile2.ReleaseNotes

Bravo notes offers simple yet flexible templates designed specifically around th
creation.

For the technical folks it might be interesting that the main template is based

- Markdown - becoming the de facto standard for authering technical docu
- Handlebars - allows for simple, robust yet powerful templating

```
{{!-- here we iterate over work items with the "feature"

{{#feature.workItems}}
### {{{ field 'System.Title' }}} {{!-- output the featur
```

The blue box and with the small heading shows you in the preview exactly where the highlighted content comes from. In this case the ReleaseNotes field of the Feature with ID 676.

## 4.2 Import work item content from query

### The problem

One core question when creating release notes is:

What is getting shipped with this release? And what do we want to talk about in the release notes?

It might be less obvious than it initially sounds!

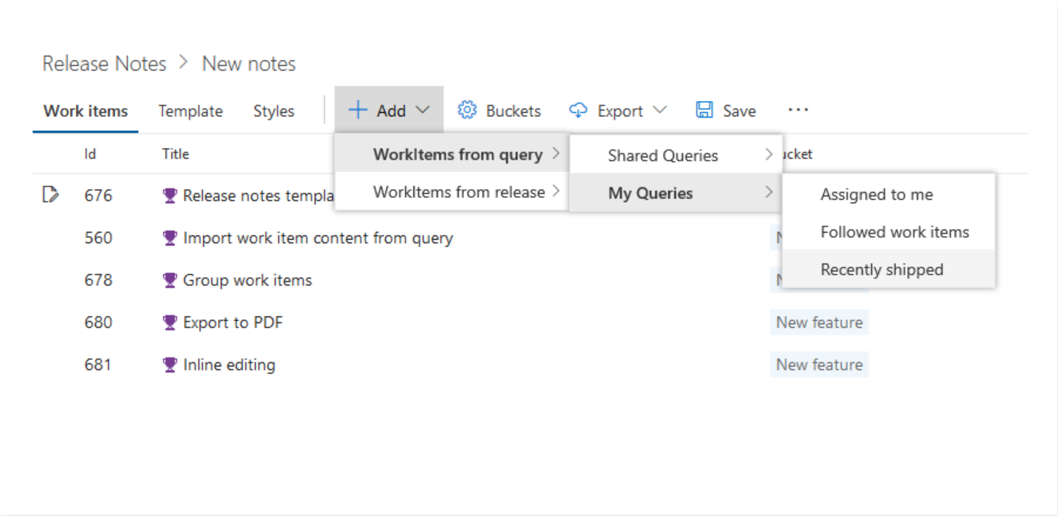
Not every feature or improvement is worth talking about. Also listing many less important items can distract from the most valuable ones.

### Our solution

Luckily VSTS and TFS have powerful querying capabilities. To separate items that we want to include or not include in our release notes we can use a query to pull in:

- Items completed in a certain iteration which are
- of a certain work item type like Feature and
- have a custom field "Requires Release Notes" toggled on and
- have the state Closed

In this release you can select any work item query and the results will be added to the work items tab of the extension.



| Id | Title | ıcket |
|---|---|---|
| 676 | Release notes templa | |
| 560 | Import work item content from query | N |
| 678 | Group work items | N |
| 680 | Export to PDF | New feature |
| 681 | Inline editing | New feature |

## 4.3 Group work items

### The problem

When looking at release notes we usually see two types of items

- New features
- Fixed bugs

Obviously we want a simple way to organize our work items in those 2 categories / groups.

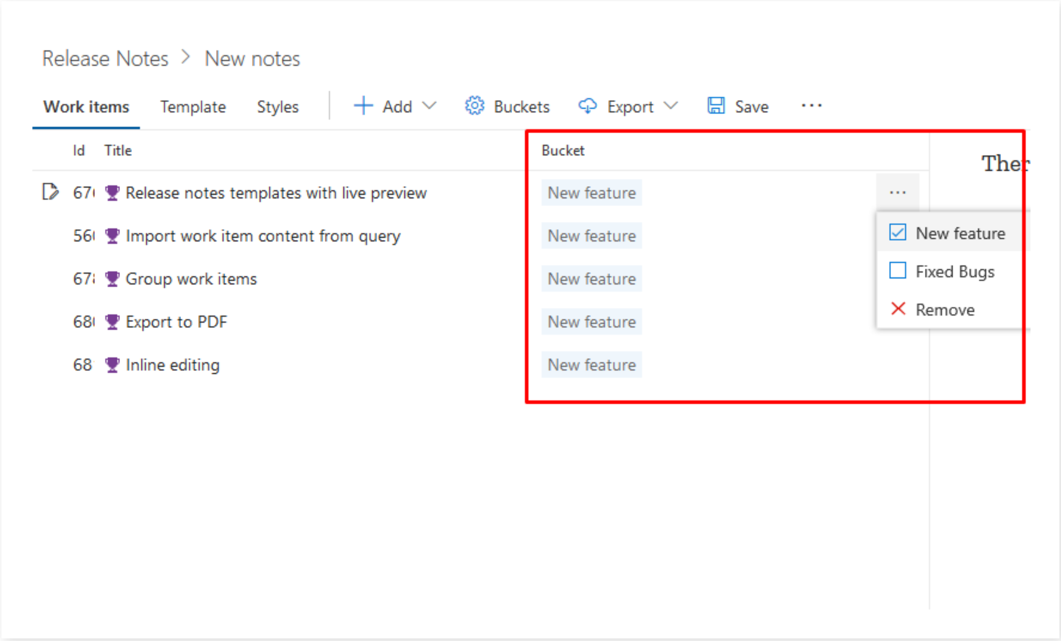
But in many cases more granular grouping is need such as:

- Items grouped by application (is a new feature part of the web app or mobile app?)
- Items grouped by area (like Work, Code, Build&Release in VSTS)
- Items grouped by importance (highlights, performance improvements)

There are probalby many more scenarios where grouping items in a certain way makes sense.

### Our solution

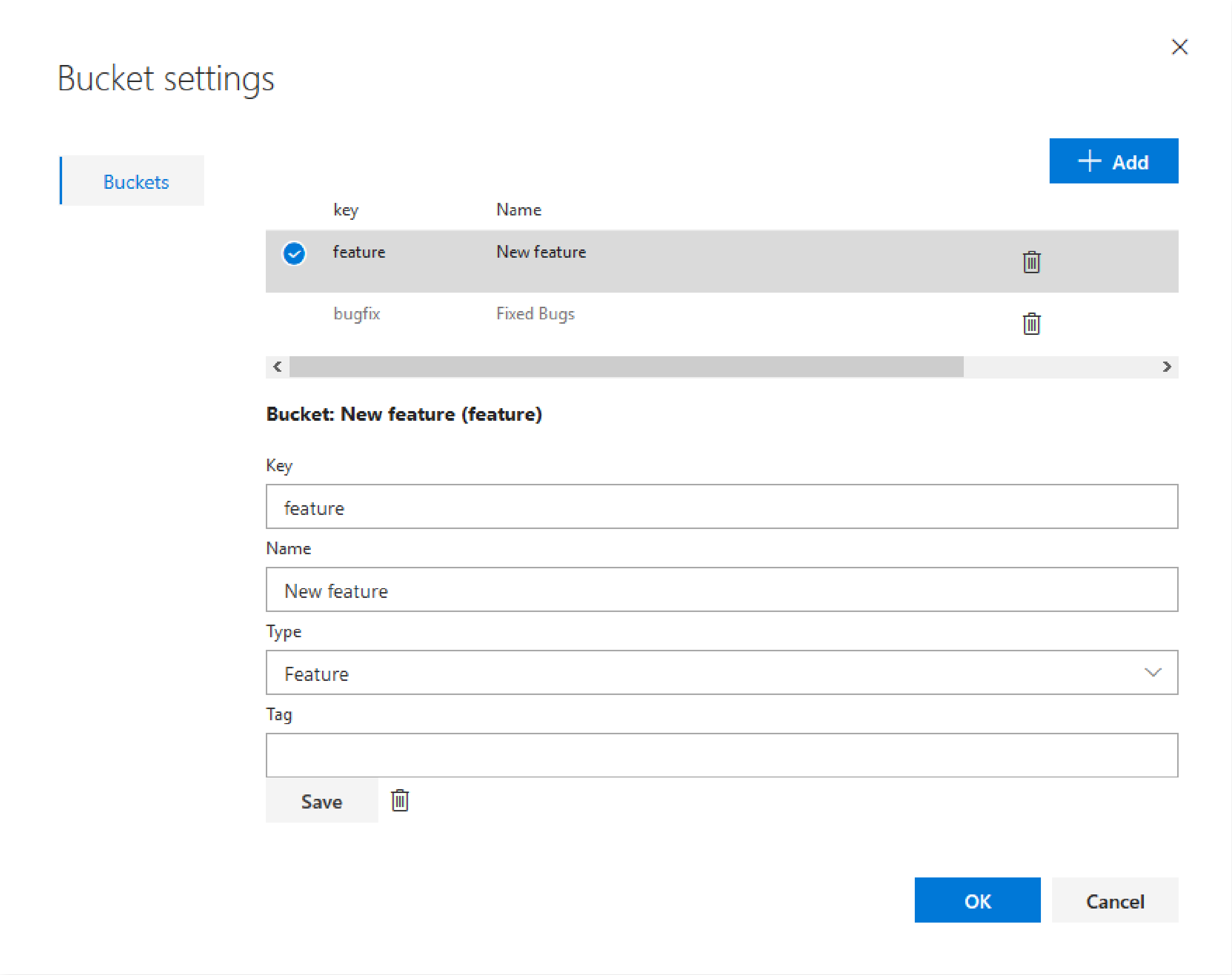
To allow flexible grouping each item in the work items tab can be assigned a "bucket" (we will probalby change the name from buckets to labels).



| Id | Title | Bucket |
|---|---|---|
| 67( | Release notes templates with live preview | New feature |
| 56( | Import work item content from query | New feature |
| 67: | Group work items | New feature |
| 68( | Export to PDF | New feature |
| 68 | Inline editing | New feature |

You are free to add whatever buckets / labels you need. For each bucket / label you can configure how it shall be applied when importing work items. Today you can automatically assign ab bucket / label

- based on the type of work item
- based on a specific tag



| key | Name |
|---|---|
| feature | New feature |
| bugfix | Fixed Bugs |

Bucket: New feature (feature)

Key: feature

Name: New feature

Type: Feature

Tag:

We will also add the ability to assign a bucket / label based on the Area Path of work items.

## 4.4 Export to PDF

### The problem

Yes I know: OMG save the trees! Why use PDF in this day and age?

PDFs seem antiquated and misplaced in many cases (isn't it just so much more fun to read PDFs than mobile optimized html on a smartphone screen?)

Still many software companies currently use PDF as a standard format to send their release notes to customers and users.
As most of the release notes content comes from VSTS / TFS - a web application - there is quite a gap to bridge to produce beautiful PDFs.

Lot of hours are wasted every release by exporting data to Excel and Word, pasting the content in a printable layout. This is definitely I pain point we need to address.

### Our solution

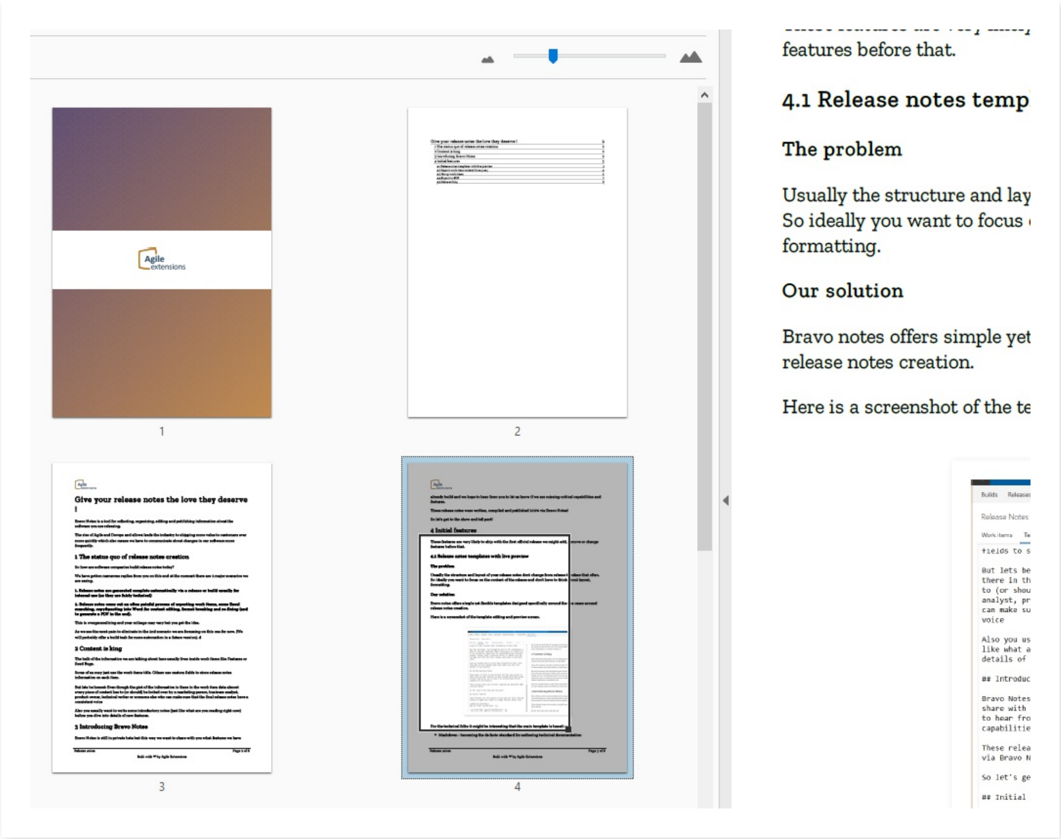
Not to pad ourselves on the back too much but for a inital release we added quite a number of features around generating PDFs.

We wanted to make sure that people are able to generate a PDF from their release notes with one click without all the intermediate steps that still looks as good or maybe even better than the documents they currently produce.

Here are the capabilities of the current PDF-Export feature:

- Layout and styles with HTML and CSS
- Support for any web font
- Individual cover page
- Table of contents based on headings
- Page header
- Page footer
- Use page numbers, section / subsection in header or footer
- Seamless embedding of images

We realize that professional word processors have a whole lot of more features to get the perfect layout. We follow the 80/20 rule here though and leveraging the Web- and CSS-skills of your developers and designers you'd be surprised how well documents generated from HTML and CSS can look like in PDF-Form!



features before that.

4.1 Release notes temp

The problem

Usually the structure and lay
So ideally you want to focus
formatting.

Our solution

Bravo notes offers simple yet
release notes creation.

Here is a screenshot of the te

```
fields to s

But lets be
there in th
to (or shou
analyst, pr
can make su
voice

Also you us
like what a
details of

## Introduc

Bravo Notes
share with
to hear fro
capabilitie

These relea
via Bravo N

So let's ge

## Initial
```

To give you an idea you can have a look at the PDF-Version of the release notes you are currently reading. Produced of course with the same features we are talking about here. (Please come back to continue reading here though)

IMPORTANT

Generating PDF documents happens on our own backend hosted securely in Azure. If data about your release notes should under no circumstances leave the security boundaries of your VSTS account / TFS server please drop us a line to discuss options for self hosting this part of the product.

## 4.5 Inline editing

### The problem

Release notes usually include information about multiple new features or other changes. Having most of the content relate directly to work items has a great benefit:

You can easily distrubute the effort of creating release notes by writing about one item at a time or have multiple people do it.

However once you bring all the content together and see a preview of the full compiled document this is also the time where you see things that need changing.

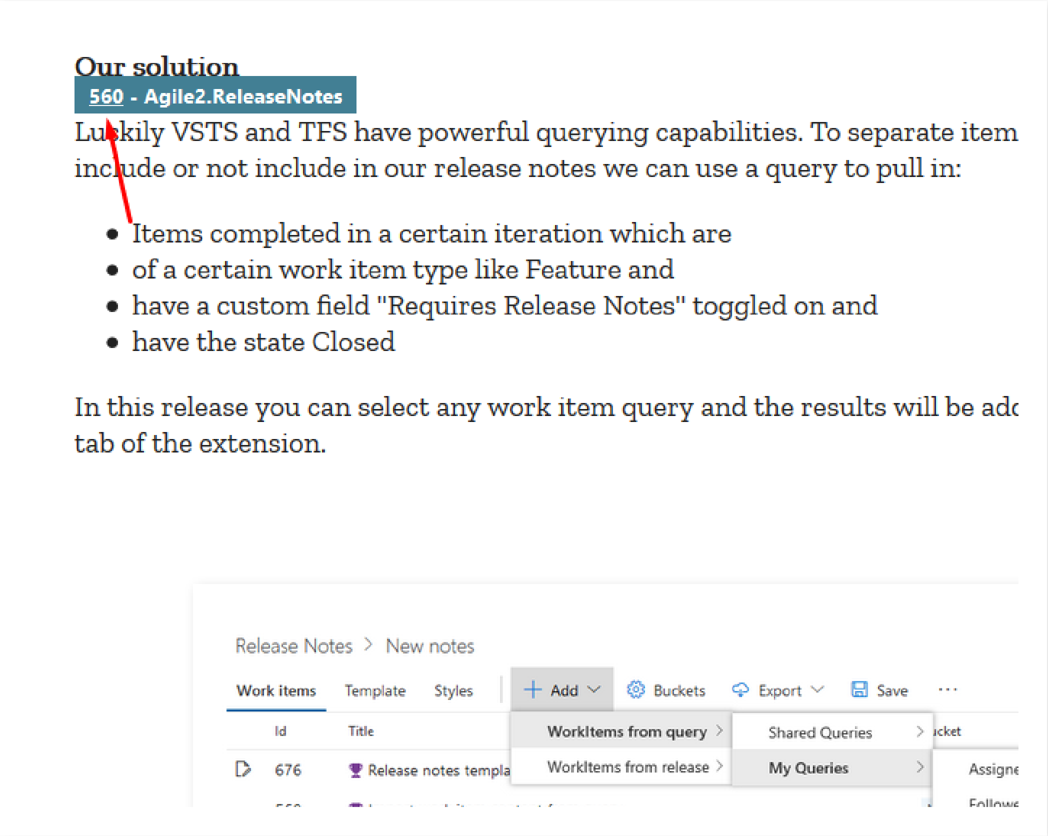
To make those changes, big or small it can be annoying to: find the work item the content is coming from, navigate to the work item, make the change, regenerate document and see how the change looks in the final document.

### Our solution

To tackle this problem we enabled two features:

1. Quickly make work item changes via the work item form
2. Inline editing of content

For the first one we allow you to just hover over a piece of content in the release notes preview and click the work item ID thats shown in the overlay:



Our solution

Luckily VSTS and TFS have powerful querying capabilities. To separate item include or not include in our release notes we can use a query to pull in:

- Items completed in a certain iteration which are
- of a certain work item type like Feature and
- have a custom field "Requires Release Notes" toggled on and
- have the state Closed

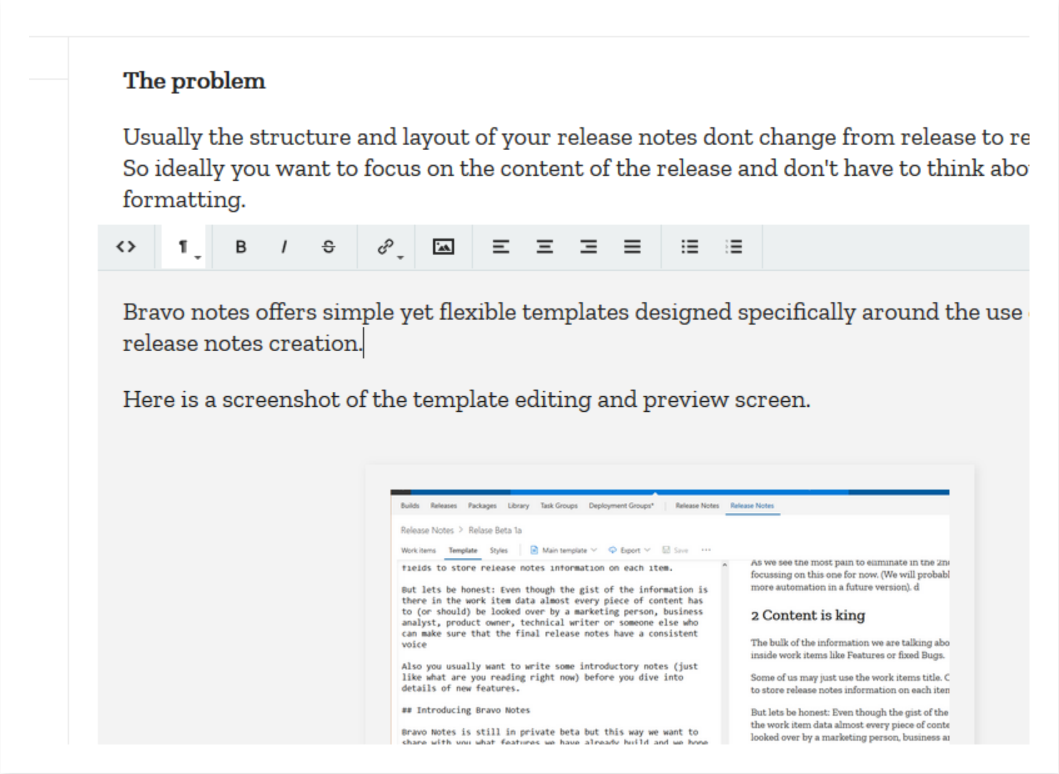
In this release you can select any work item query and the results will be ad tab of the extension.

After saving and closing the work item form the changes will be immediately applied to your release notes and are visible in the preview instantly.

The second appraoch of inline editing is still in an experimental state but already works pretty well.

Inline editing allows you to just click into a piece of content in the preview and make edits right there!

There are still some issues with the inline editor but our goal is to make the editing experience so smooth you never want to open a work item form to edit release notes content ever again!



**The problem**

Usually the structure and layout of your release notes dont change from release to re
So ideally you want to focus on the content of the release and don't have to think abo
formatting.

Bravo notes offers simple yet flexible templates designed specifically around the use
release notes creation.

Here is a screenshot of the template editing and preview screen.

```
fields to store release notes information on each item.

But lets be honest: Even though the gist of the information is
there in the work item data almost every piece of content has
to (or should) be looked over by a marketing person, business
analyst, product owner, technical writer or someone else who
can make sure that the final release notes have a consistent
voice

Also you usually want to write some introductory notes (just
like what are you reading right now) before you dive into
details of new features.

## Introducing Bravo Notes

Bravo Notes is still in private beta but this way we want to
share with you what features we have already build and we hope
```

As we see the most pain to eliminate in the 2n
focussing on this one for now. (We will probabl
more automation in a future version). d

2 Content is king

The bulk of the information we are talking abo
inside work items like Features or fixed Bugs.

Some of us may just use the work items title. C
to store release notes information on each iten

But lets be honest: Even though the gist of the
the work item data almost every piece of conte
looked over by a marketing person, business a

# 5 What's next?

The release notes you have just read are the first time the product has generated value. As the first customer I have to say I am pretty happy considering its the very first version!

Having the ability to create content on a work item level and still get immedidate feedback on what the content will look like in the final document is definitely an improvement. Also the template made it really easy to focus on the content and forget about the structure once it is set up.

It is time now though to create some release notes for products that are already in the market. We have selected a few customers to join our private beta and start using the product for their release notes.

Before diving into building new features we want to make sure that the ones we built are rock solid and save you as much time as possible.

# 6 Want to join the private beta?

If you like what you see above and are interested in joining the private beta please let us know via hello@agileextensions.com.