



Management Server internals

Pagebuilder, Publication, Clean up live server

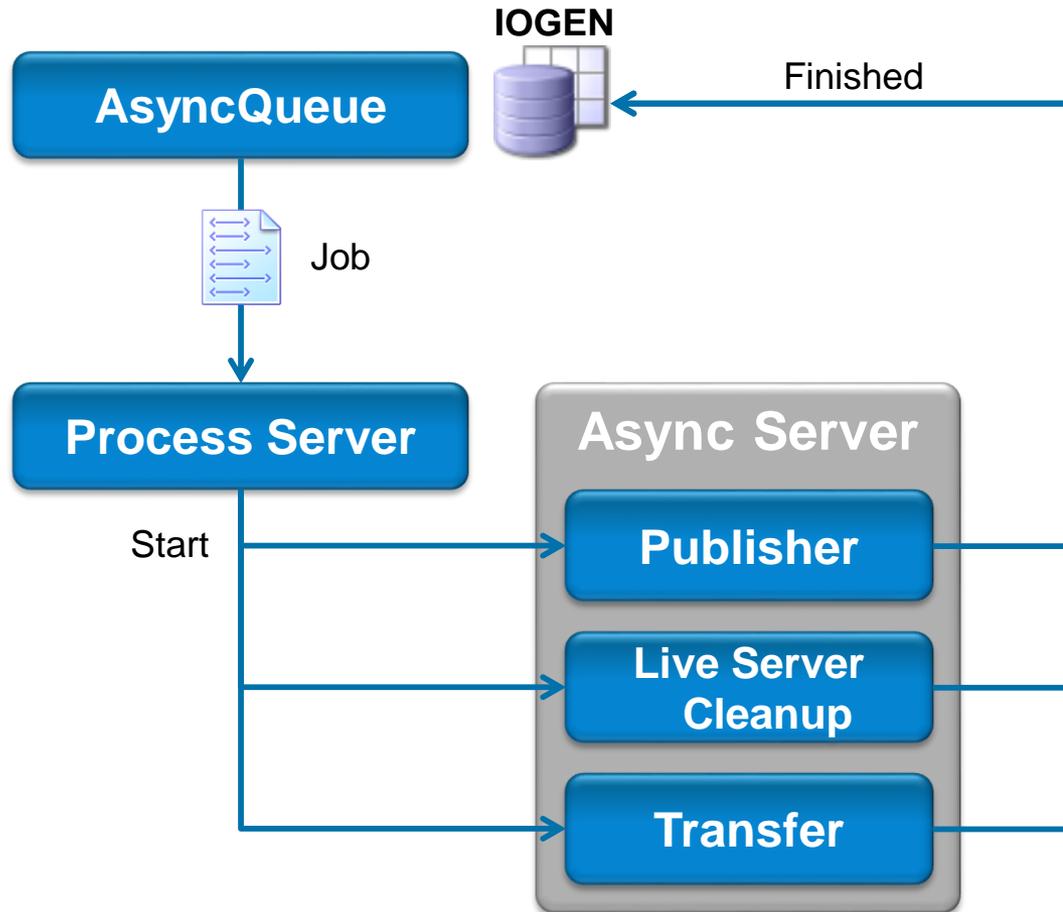
Martin Richter
Hamburg, May 18th, 2015

Agenda

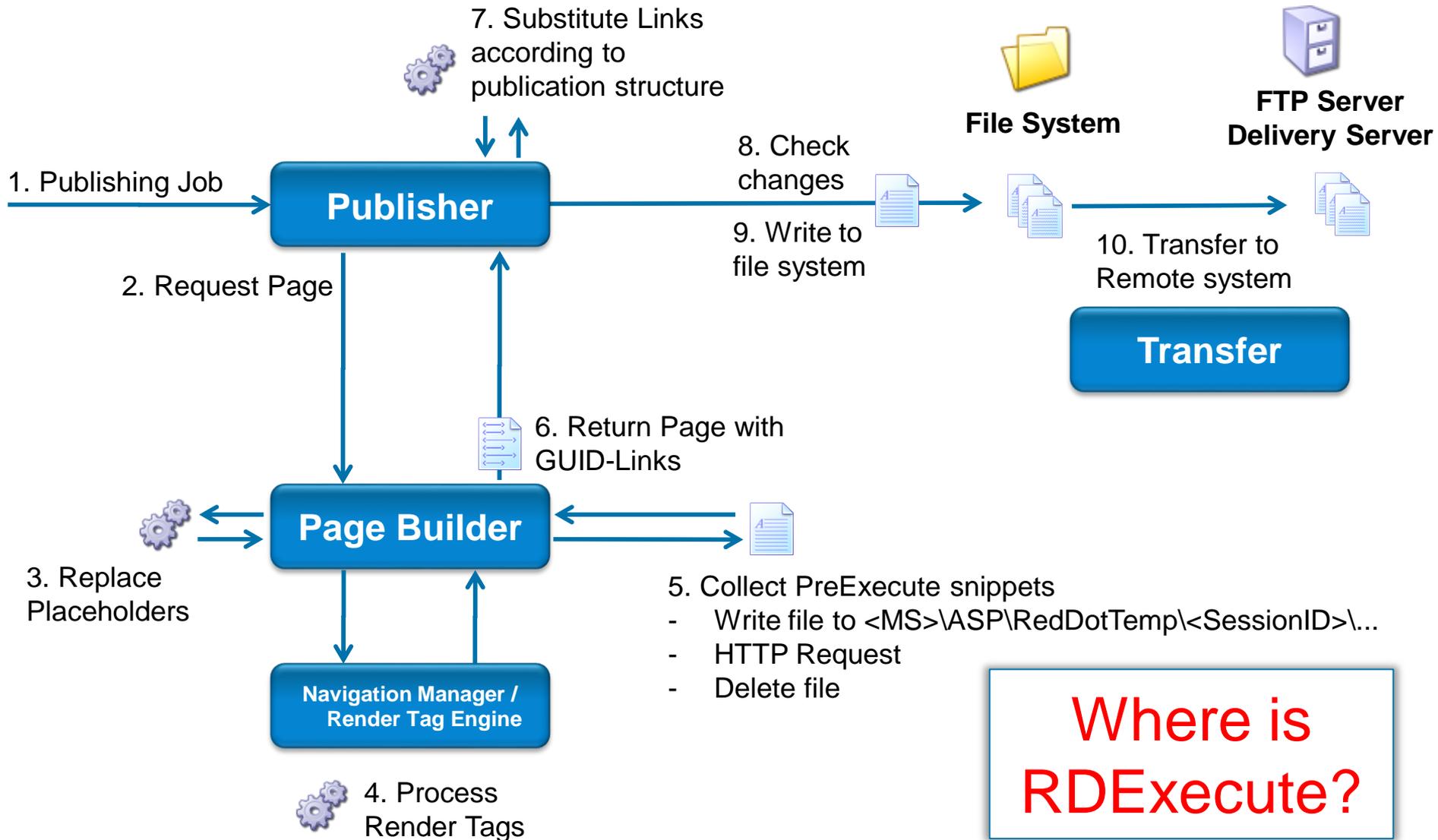
- Publish Pages
 - Publisher, PageBuilder, Transfer
 - Pages, Render Tags, Pre-/RDExecute, Elements
- Delete Pages
 - Garbage collection, workflow delete, orphaned pages
- Related Pages
- PreExecute / RDExecute
 - Overview, Debugging

Publish Pages

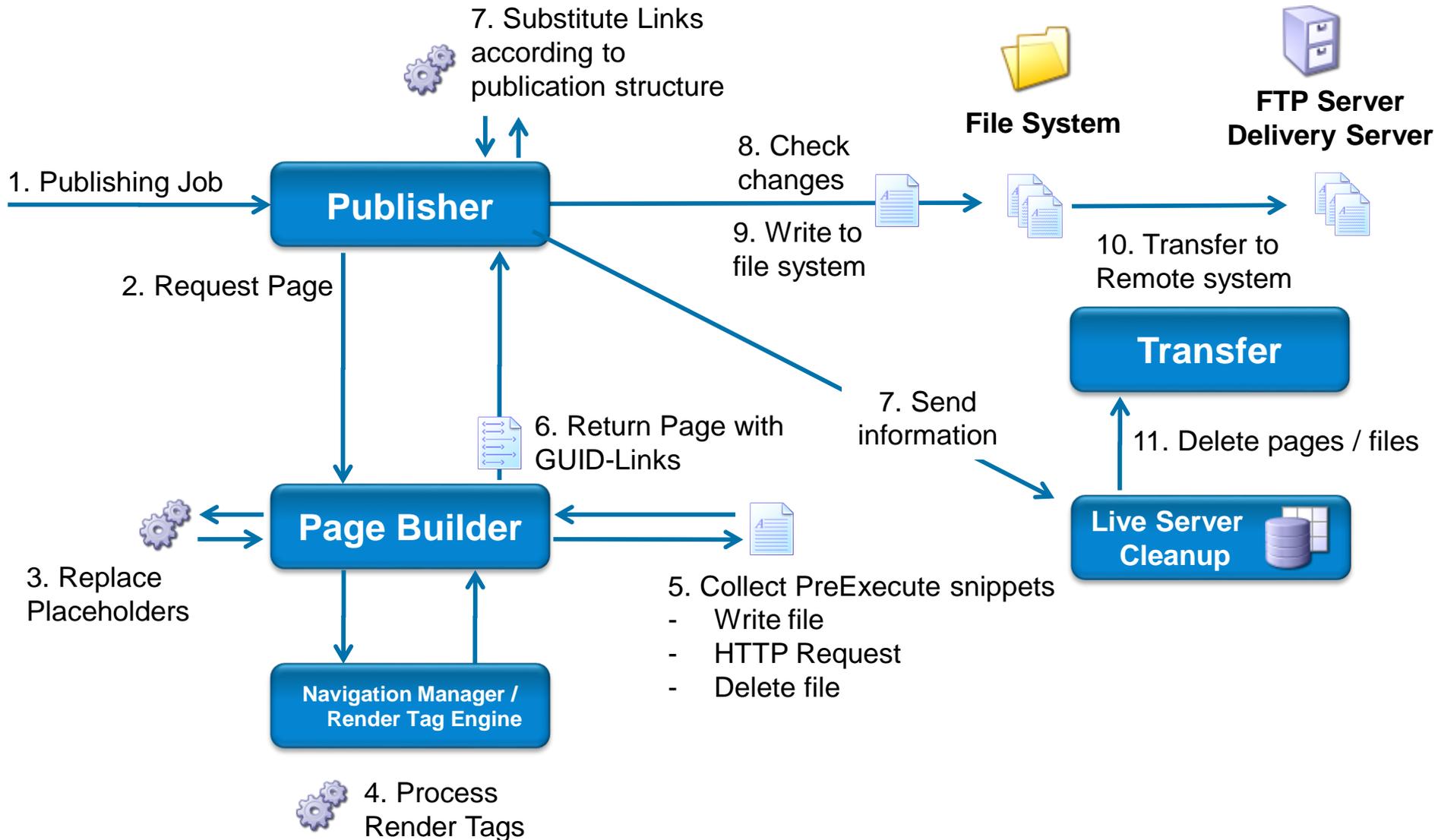
Publish Page



Publish Page

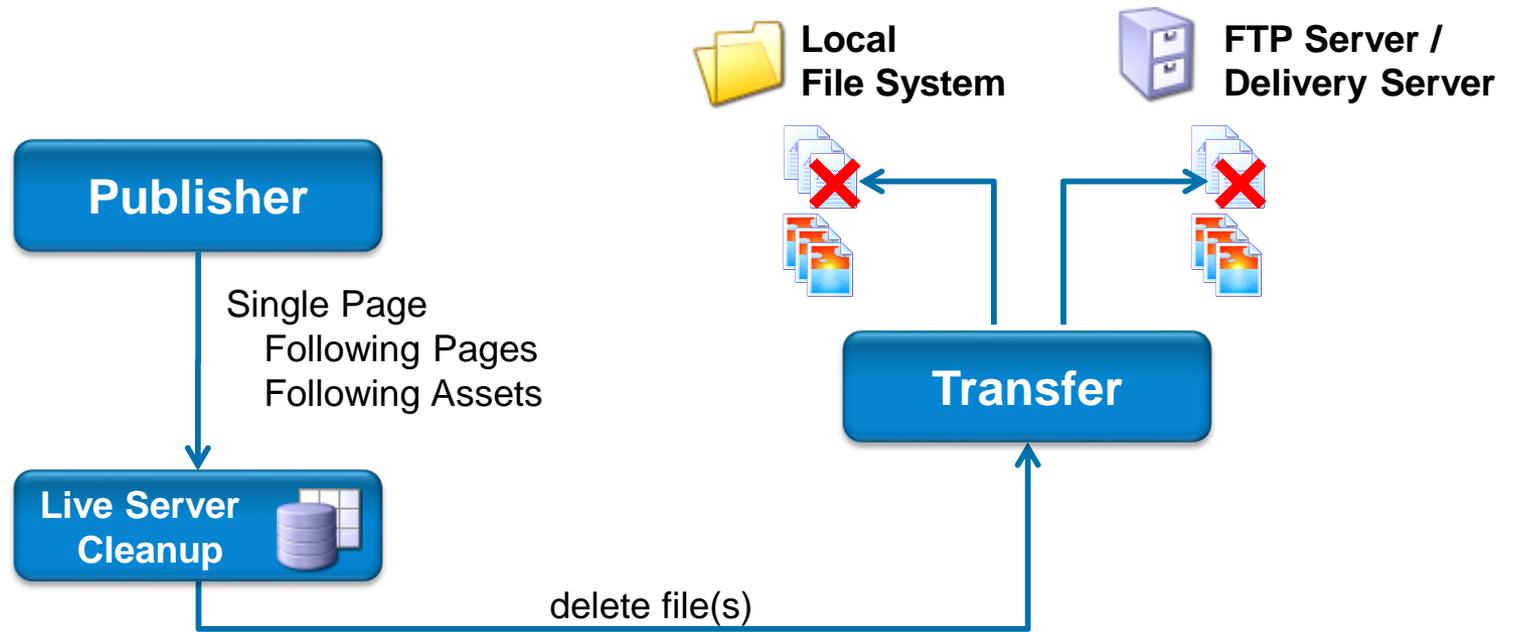


Publish Page



Delete Pages

Delete Page



Publisher – collect information for cleaner index

- During publishing of a page
 - All relevant information for page is gathered (file name, publishing target, path, language and project variant ...)
 - All following pages are gathered (only basic information like page guid, link, ...)
 - All assets are gathered
- Information is written in a file in RedDotTemp folder
- Information is xml style:

```
<PAGEINFO pageguid="057239F96D5A4EDAAC98A2691E5F1E63">  
  <LINKINFO></LINKINFO>  
  <IMAGEINFO></IMAGEINFO>  
</PAGEINFO>
```

Cleaner – process information from publisher

- Cleaner is scanning files in RedDotTemp folder
- Live server cleaning index will be updated:
 - No longer existing connections will be removed
 - Pages/assets now no longer connected will be marked
 - New pages and connections will be added
 - Page delete action removes all connections to the deleted page
- Marked pages/assets will be checked:
 - If still not connected, entry will be deleted and page/asset is marked for deletion
 - Following pages/assets of pages marked for deletion will be checked recursive

Transfer – Handle files on live server

- Pages/assets that are marked for deletion will be removed in RedDotTemp if publishing target is DS or FTP
- Pages/assets that are marked for deletion will be removed directly on live system if publishing target is local or UNC
- If file name in DS project is unique within project and only group has been changed, file is not deleted from DS
- In all other cases, delete list will be created for each DS group containing deleted files and send to DS

Cleanup strategies

Two approaches

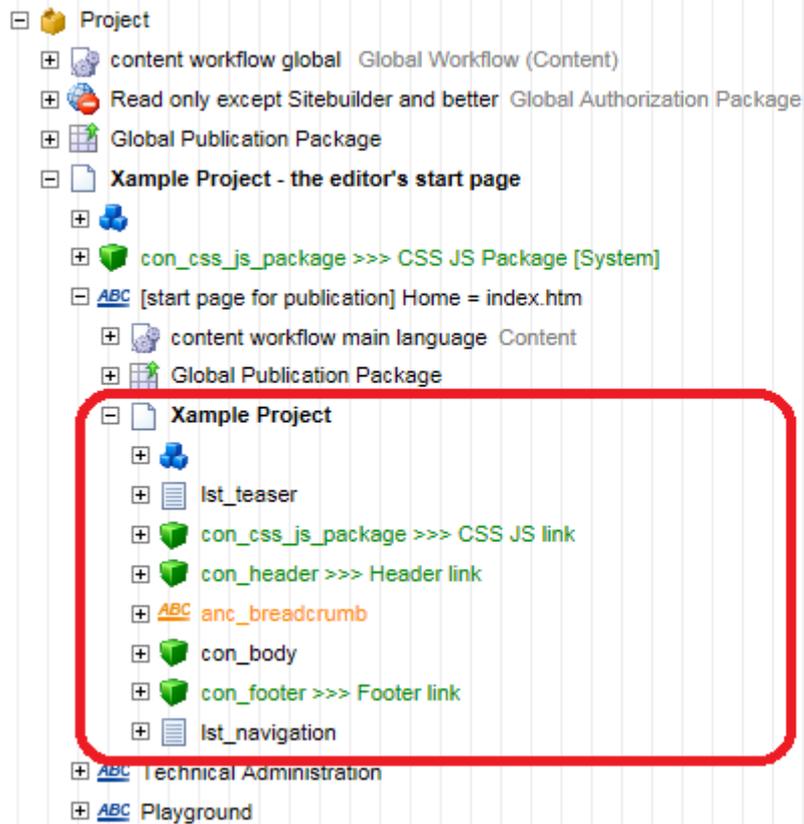
1. Garbage collection approach
 - Remove files if you are sure that they are not used anymore
2. Workflow approach
 - Delete first ask later

Garbage Collection Approach

Publishing has two phases

1. Publish pages = write files
 - Cleaner index registers all published pages
 - Cleaner index registers all references
2. Cleanup = remove files
 - Cleaner calculates to be deleted files

No liveserver cleanup for subtree publishing



Liveserver cleanup begins at root page

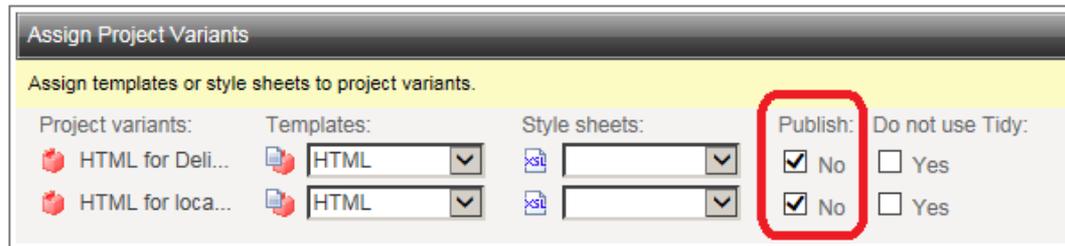
- Project
 - content workflow global Global Workflow (Content)
 - Read only except Sitebuilder and better Global Authorization Package
 - Global Publication Package

start publication here

- Xample Project - the editor's start page
 - con_css_js_package >>> CSS JS Package [System]
 - ABC [start page for publication] Home = index.htm
 - content workflow main language Content
 - Global Publication Package
 - Xample Project
 - lst_teaser
 - con_css_js_package >>> CSS JS link
 - con_header >>> Header link
 - anc_breadcrumb
 - con_body
 - con_footer >>> Footer link
 - lst_navigation
 - ABC Technical Administration
 - Global Publication Package
 - Technical Administration
 - con_css_js_package >>> CSS JS Package [System]
 - Shared Components
 - CSS
 - JS
 - Single Pages Publish Separately
 - ABC Playground
 - Global Publication Package
 - Playground
 - con_css_js_package >>> CSS JS Package [System]
 - anc_dyn_links
 - anc_dyn_links
 - anc_dyn_links
 - anc_dyn_links

How not to publish unwanted pages

- Content Classes -> Templates -> Assign Project Variants



- Start publication at root node
- „Publish No“-following pages will be published
- Eliminate/Hide unwanted links for publishing
 - example:

```
<reddot:cms>
  <if>
    <query valuea="Context:CurrentRenderMode" operator="!=" valueb="Int:2"><htmltext>
      <div class="content">
        <a href="<%anc_dyn_links%"></a>
      </div>
    </htmltext></query>
  </if>
</reddot:cms>
```

Liveserver cleanup begins at root page

- Project
 - content workflow global Global Workflow (Content)
 - Read only except Sitebuilder and better Global Authorization Package
 - Global Publication Package

Xample Project - the editor's start page

- con_css_js_package >>> CSS JS Package [System]
- ABC [start page for publication] Home = index.htm
 - content workflow main language Content
 - Global Publication Package

do not publish

Xample Project

- lst_teaser
- con_css_js_package >>> CSS JS link
- con_header >>> Header link
- anc_breadcrumb
- con_body
- con_footer >>> Footer link
- lst_navigation

publish

Technical Administration

- Global Publication Package
- Technical Administration
 - con_css_js_package >>> CSS JS Package [System]
 - Shared Components
 - CSS
 - JS
 - Single Pages Publish Separately

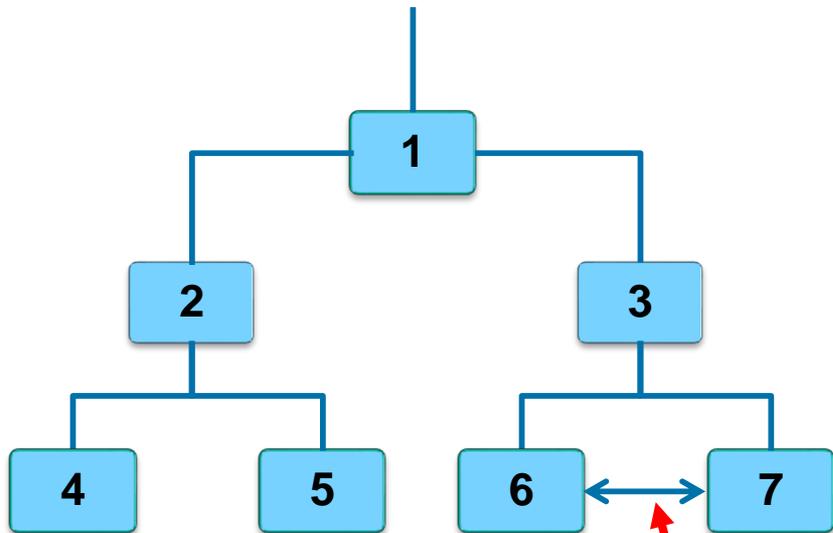
do not publish

Playground

- Global Publication Package
- Playground
 - con_css_js_package >>> CSS JS Package [System]
 - anc_dyn_links
 - anc_dyn_links
 - anc_dyn_links
 - anc_dyn_links

How to create orphaned pages

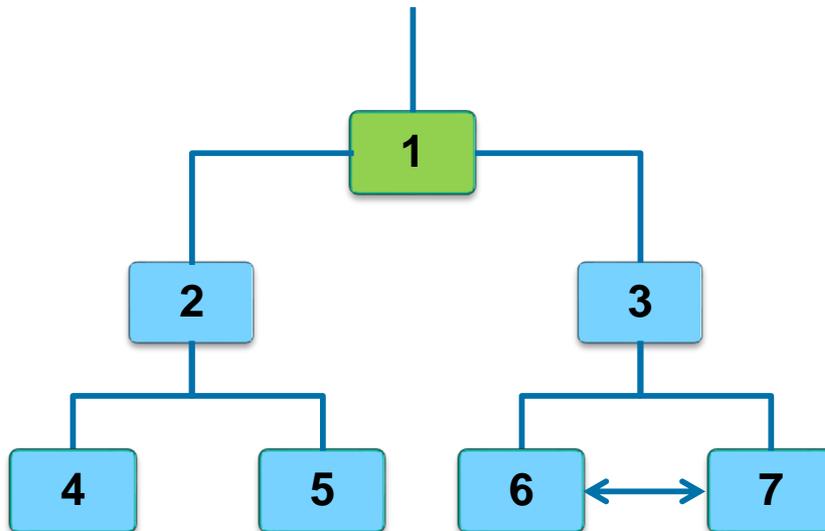
- Structure with additional cross-links



Connections between pages

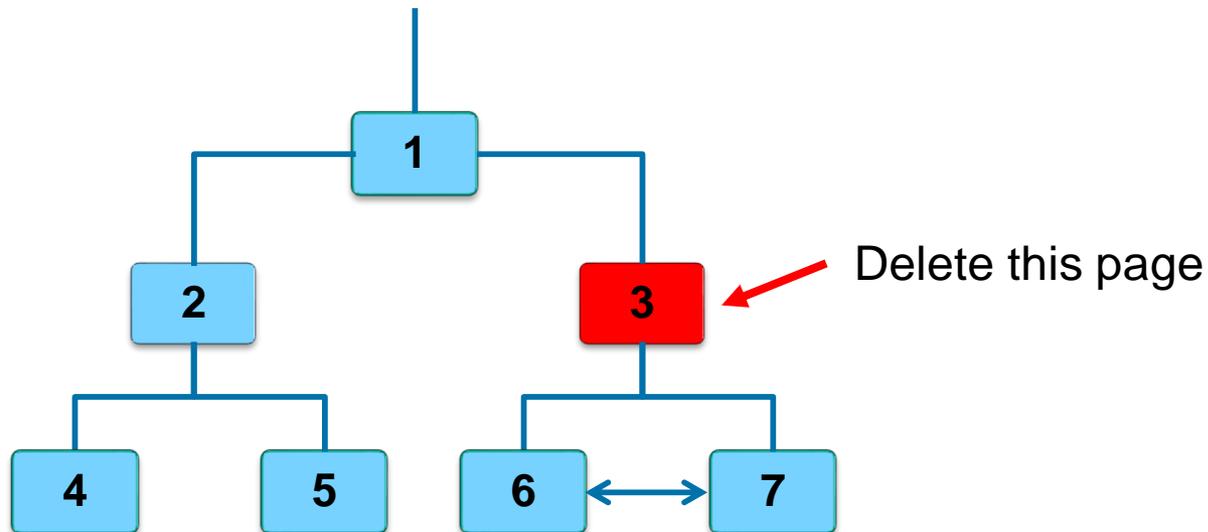
How to create orphaned pages

- Publish with following pages



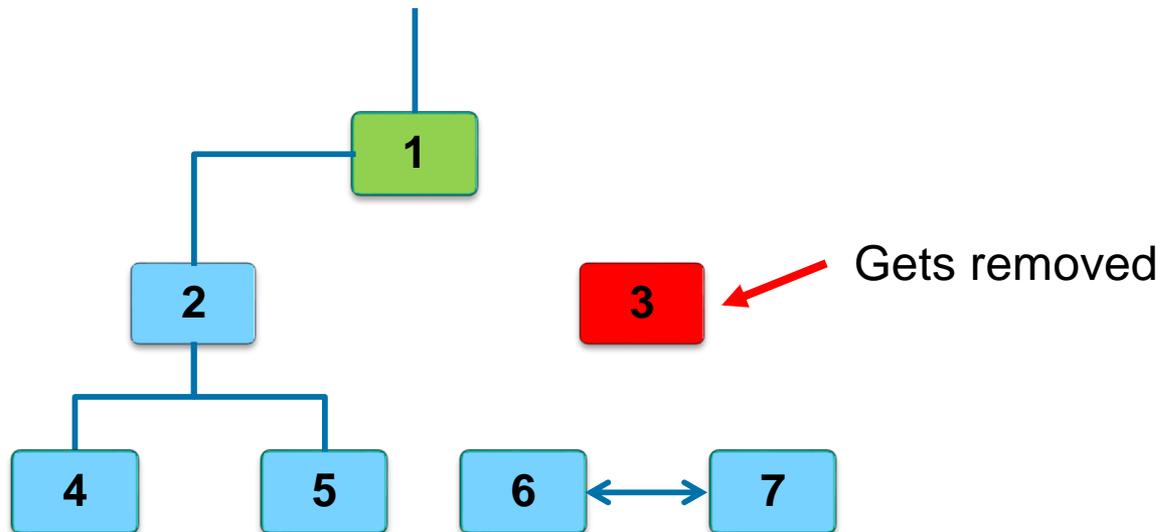
How to create orphaned pages

- Delete a page so that a substructure remains



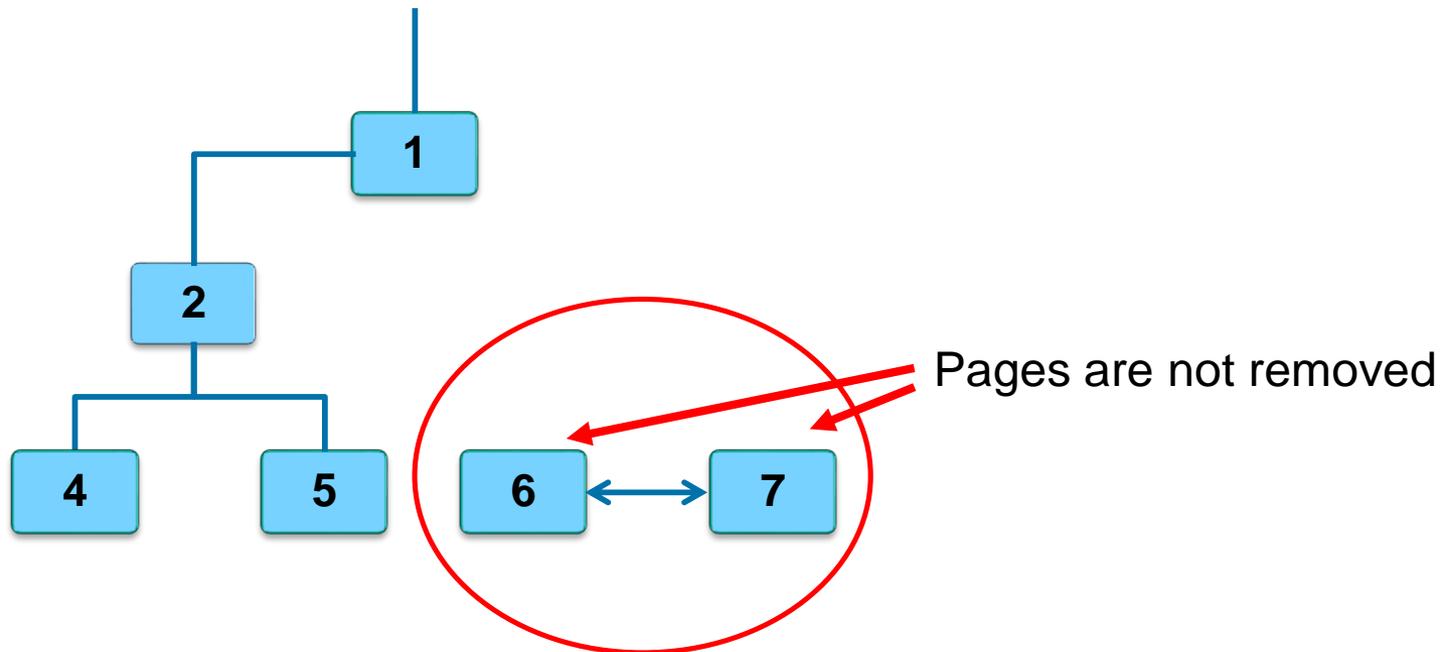
How to create orphaned pages

- Publish with following pages



How to create orphaned pages

- Interconnected substructure is not deleted!



How to remove orphaned pages

- Purge orphans with CLEANER action="delete"
 - <PROJECT>
 - <CLEANER action="delete"
 - projectvariantguid="[PROJECT_VARIANT_GUID]"
 - languagevariantguid="[LANGUAGE_VARIANT_GUID]">
 - <EMAIL subject="[MAIL_SUBJECT]" userguid="[!guid_user!]">
 - <![CDATA[You can insert an optional email text here.]]>
 - </EMAIL>
 - </CLEANER>
 - </PROJECT>

How to detect orphaned pages (“Dry Run”)

- Detect orphans with CLEANER action=“list“
 - <PROJECT>
 - <CLEANER action=“list“
 - projectvariantguid=“[PROJECT_VARIANT_GUID]“
 - languagevariantguid=“[LANGUAGE_VARIANT_GUID]“>
 - <EMAIL subject=“[MAIL_SUBJECT]“ userguid=“[!guid_user!]“>
 - <![CDATA[You can insert an optional email text here.]]>
 - </EMAIL>
 - </CLEANER>
 - </PROJECT>

How to detect remaining connections

- Garbage collection says: If a page is not deleted, it is still used by another page
- Retrieve remaining connections from the cleaner index
 - Show connections to a page
 - `<CLEANER action="listallpagesto"
projectguid="[PROJECT_GUID]"
entryid="[page_id]" />`
 - Show connections to a page or file
 - `<CLEANER action="listallpagesto"
projectvariantguid="[PROJECT_VARIANT_GUID]"
entryid="[file_name]" />`

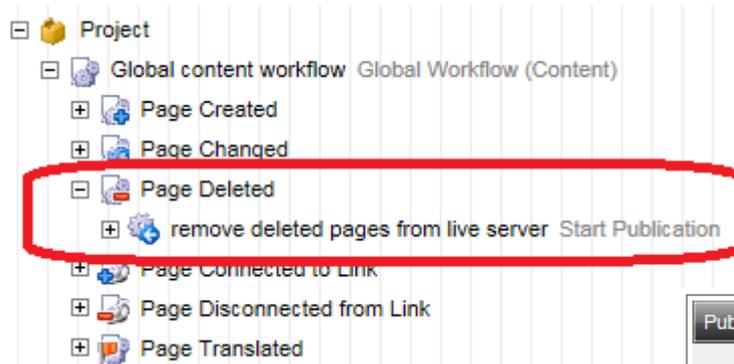
Garbage Collection Approach

Summary

- Start project publication at project root
- Set „Publish No“ for unwanted pages
- Project publication deletes unused pages
- Purge orphans with CLEANER action=“delete“

Workflow publishing

- Delete first ask later
- Workflow publication on Page Deleted



- Removes deleted pages
- Publishes related pages



Note: avoid publishing unlinked pages

- Publishing option: publish unlinked pages

Always replace media files	<input checked="" type="checkbox"/> Yes
Publish pages even if asset repositories are currently not available	<input type="checkbox"/> Yes
Publish unlinked pages	<input type="checkbox"/> Yes

- If selected:
 - free and other unlinked pages can be published.
 - published following pages of a deleted page will not be deleted from the live server.

Planning: new publication option

- Direct cleanup option
- Delete-page-publication option available without workflow
- Planned for Q3 / CY15

Related Pages

Related pages

- Related pages are all pages that
 - make use of a page
 - may be affected by changes of the page
- Unchanged related pages are not transferred to the live server

Related pages

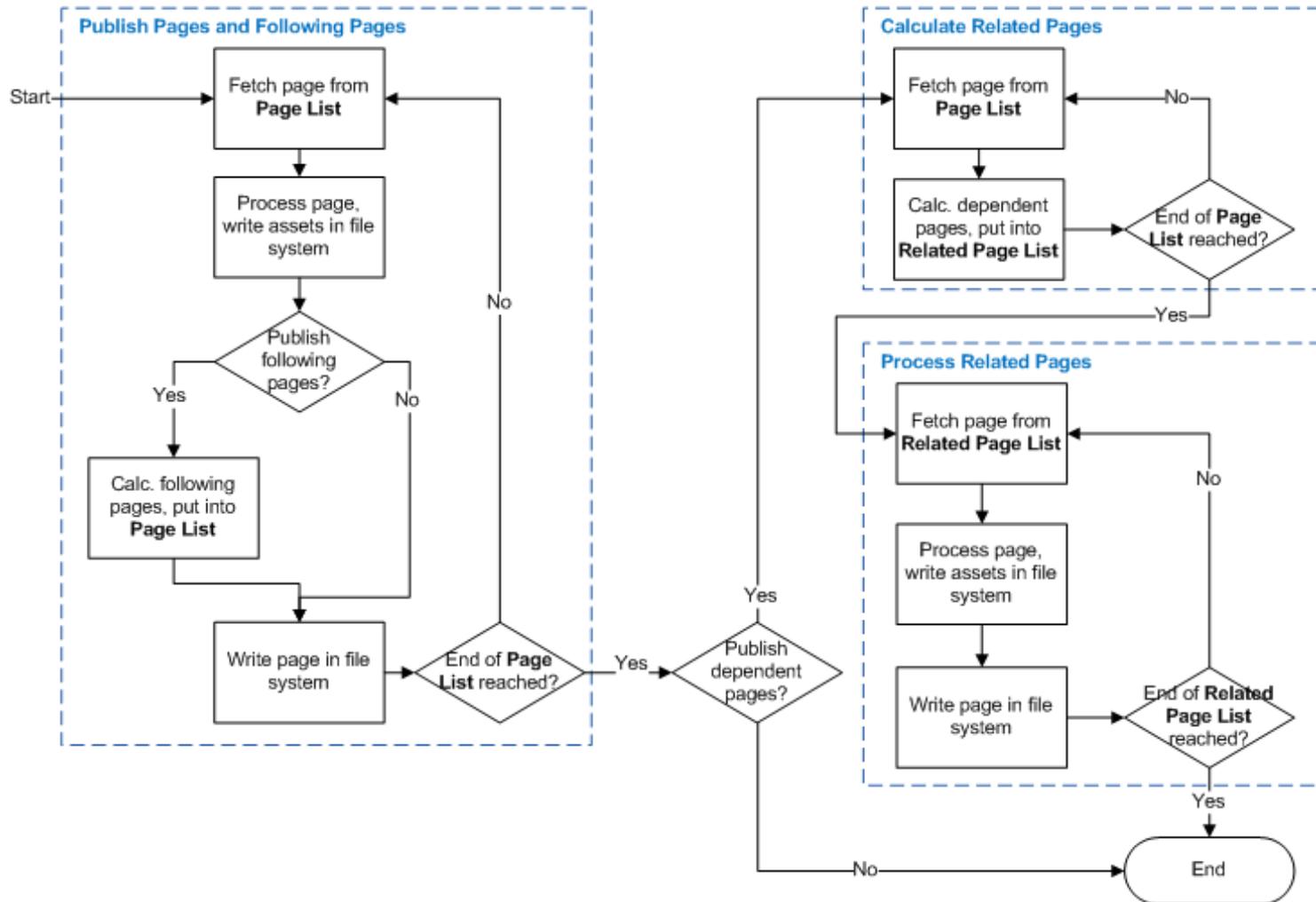
- Checked:
 - Project structure
 - Links
 - References
- Checked as good as possible:
 - Template code
- Not checked:
 - Links in text elements
 - Project content element
 - Reference page from other project
 - Navigation templates
 - Render tags

Options

- Add Cleaner index information to related pages list
 - RDServer.ini
 - [PUBLISHER]
 - checkIsdatabase=1
- Template code check
 - Always update for changes
 - Project option

Use pessimistic approach for checking related pages

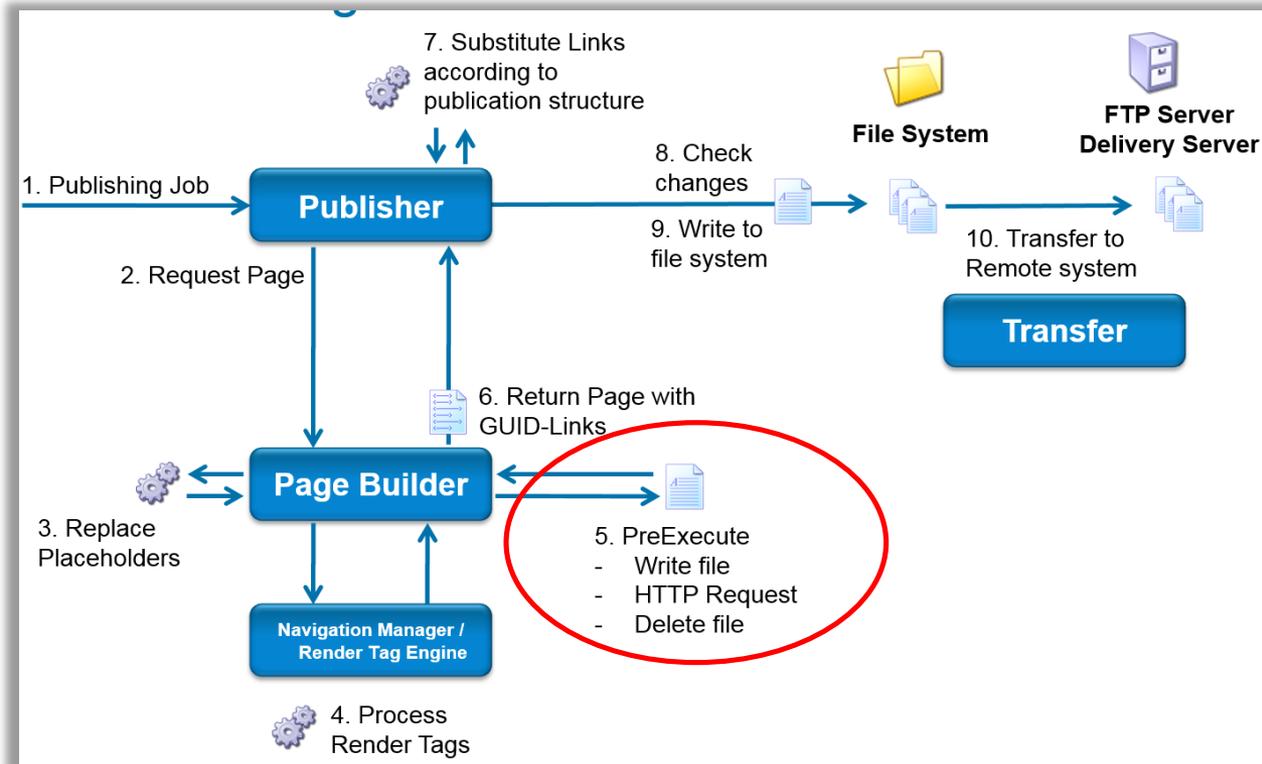
Publishing pages and related pages



PreExecute / RDExecute

Overview PreExecute / RDExecute

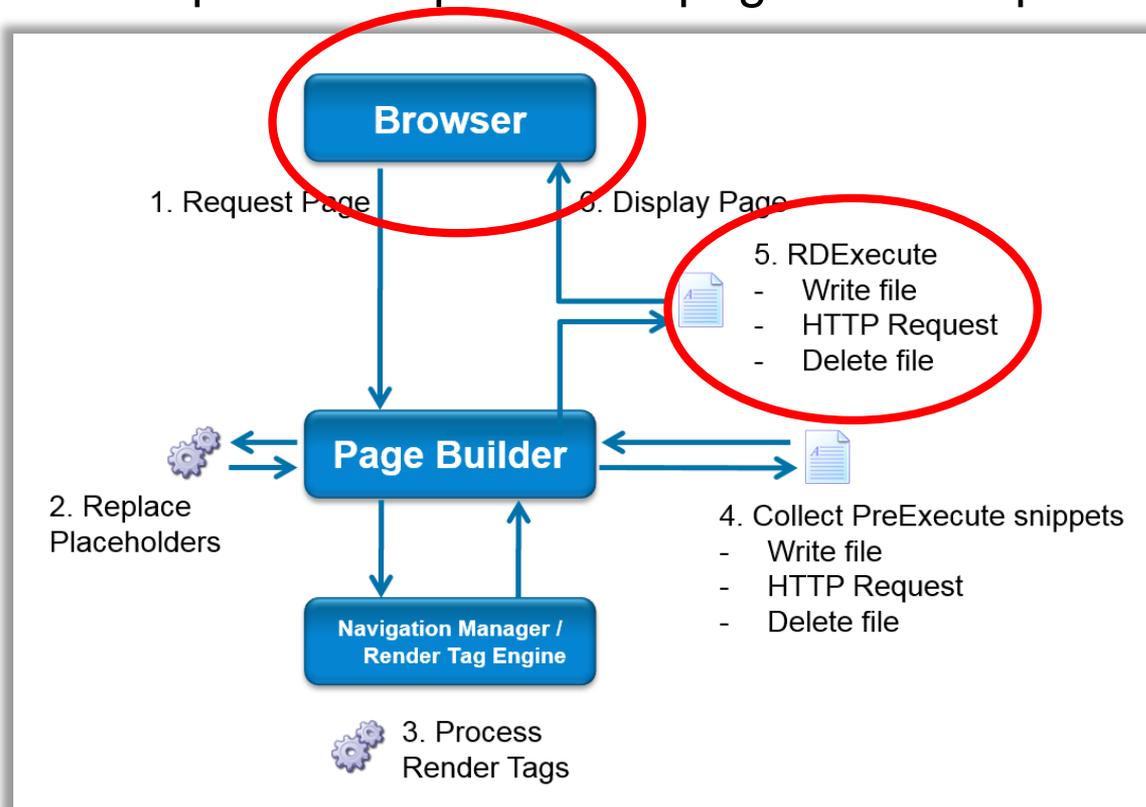
- PreExecute
 - Process script code before a page is published



Overview PreExecute / RDExecute

- RDExecute

- Have correct preview if published pages are script code pages



Overview PreExecute / RDExecute

- PreExecute
 - Usage: Process script code before a page is published
 - Used in Active Templates
 - Publish the result of the processed script code
 - Caching will be prohibited by
 - RedDots
 - Block marks
 - Render mode evaluation
 - Several types of Info elements (date/time, ...)
- RDExecute
 - Usage: Have correct preview if published pages are script code pages
 - Process the whole page
 - Displays the results in SmartEdit and page preview
- RDExecute and PreExecute in one template
 - same or different script languages possible

Enable PreExecute / RDEExecute debugging

- When running a PreExecute or a RDEExecute page, a temporary file is created in
<MS>\ASP\RedDotTemp\<SessionID>\...
- The file is deleted immediately after completion
- To debug - set the 2⁸ bit to the flags attribute in RDSERVER.INI file
[Defaults]
flags=256
- RDEExecute: Flag is cached in ASP session...

Debug PreExecute / RDEExecute

- Call PreExecute or RDEExecute page in preview or SmartEdit
- Folder <MS>\ASP\RedDotTemp\<<SessionID>
- PreExecute files:
PreExecute_<preview_call_guid>_<project_guid>_<page_guid>_temp.<ext>
- RDEExecute files:
rdexecutetemp_se_<preview_call_guid>_<project_guid>_<page_guid>_temp.<ext>
- Call PreExecute or RDEExecute page manually in the browser

OPENTEXT™

www.opentext.com



twitter.com/opentext



facebook.com/opentext



linkedin.com/company/opentext