Known Limitations

A known limitation is a limitation of the designed and implemented product. Unsupported functionality is not a limitation. The known limitations section of the release notes provides brief descriptions and workarounds to high priority and high severity product limitations.

Important: Limitations can be difficult to write. Although you might write just a few sentences, you must be fully familiar with the product and the circumstances that surround the limitation before you can convey it accurately and succinctly in the Release Note writeup. Do not rely on terminology in the bug tracking system, as much of the terminology is internal. Verify that you use terminology that is consistent with user doc and the product.

Known Limitations Criteria

Before you document a known limitation, verify that the limitation meets criteria for the release notes. Informatica includes high priority and high severity product limitations that Development is committed to fixing.

When you review known limitations that are release note candidates, consider the following criteria:

Limitation type

Known limitations must be functional limitations. Do not include enhancement requests or documentation limitations. If the nature of the limitation is functional and the type is "Documentation," look for the linked functional limitation.

Component and Sub-component

Verify that the component and sub-component are not related to documentation.

Owner

Verify that Documentation or a Doc team member does not own the bug. If the bug is a functional bug and is assigned to a writer, assign the bug back to the Development team.

Priority and Severity

Include only limitations with a high priority and high severity. Priority must be P1 or P2, and severity must be S1 or S2. In some circumstances, you can include P3, S3 limitations. Do not document P4, S4 bugs.

Status

Each item in the known limitations list must be unresolved. An unresolved bug can have the status of "open," "assigned," or "submitted."

If the status is **Resolved**, verify with Dev and QA if they plan to close it before the release date.

If the status is **Closed**, let Dev and QA know that you cannot document this as a known limitation. Sometimes they will close it with a note that Documentation is going to write it up. If they are not going to fix it, you can put it in a KB or in user doc, based on the severity of the issue. Let the Dev team know that you cannot include closed CRs as known limitations in the release notes.

Content to include

After you verify the limitation type, priority, and status, review the content of the limitation. Include the following types of errors in the known limitations:

- Product failures, such as hanging, termination, or error response to user request
- System failures, such as core dumps and crashing
- Inconsistent data or unexpected results
- Regressions
- Any other issue that might result in a call to Informatica Global Customer Support

Content to exclude

Do not include the following types of bugs in known limitations:

- Any limitation that might affect install or upgrade. For example, the installation fails with memory errors, or the upgrade fails to upgrade privileges properly. Document all install and upgrade limitations in the Installation topic of the release notes.
- PAM-related issues of supported systems. For example, do not include a limitation requesting the support of a particular version of an OS for a product.
- Corner case bugs, or bugs that are unlikely to be found by a customer. If a limitation is extremely difficult to reproduce or was possibly discovered by QA under severe stress testing, consider exclusion from the release notes.
- Requests for additional functionality. A bug that indicates a request for support of additional functionality or a feature is not considered a limitation for release notes. Key words to watch for are "support" and "should have."
- Bugs that are minor irritations. This can include misaligned user interface text, misspellings, and too many clicks or scrolls.
- Bugs that are not reproducible.

Exceptions

If you question whether to include a limitation or not, consider whether excluding the limitation from the release notes would result in a call to GCS. Note the following circumstances when we can relax the release note criteria:

- New products with a customer base that is vocal about documenting all limitations

- End-user products where limitations such as abnormal scrolling or too many clicks are higher priority
- Special requests from product management, development, or QA

Known Limitations Write-up

Write up a known limitation to describe the limitation instead of the expected behavior. Include any workaround.

A known limitation has the following elements:

Bug number

The bug number is the functional bug number in Jira. If Development linked a bug for the doc impact, do not use the documentation bug number.

Bug description

When you write a known limitation, write a short description of the bug behavior instead of the expected behavior.

- ***Change:*** The RestoreDomain command should not generate an exception if you set the -tc option.
- ***To:*** The RestoreDomain command generates an exception if you set the -tc option.

Workaround

If a limitation has a workaround, include it under the limitation. If Jira does not include a workaround, ask Dev/QA to provide one.

Do not document the following types of bugs as limitations that have the following workarounds:

Custom property

Document custom properties, called undocumented flags, as an internal KB article.

System patch

Document operating system patches in the installation section of the release notes. QA provides this information to Documentation. If the workaround is a patch that is not in the list of patches, verify with QA whether it belongs in the list. If you need to document the behavior, it might fit as a separate topic in the installation section of the release notes.

Fixed Limitations

Document fixed limitations that were reported by customers or were reported as limitations in a previous release.

When you write a fixed limitation, write a short description of the bug behavior instead of the fixed behavior.

- ***Change:*** The Column Profiling Details dialog box appears when you view the column profile for a source column in a mapping specification.
- ***To:*** The Column Profiling Details dialog box does not appear when you view the column profile for a source column in a mapping specification.

If a fixed limitation was previously documented as a known limitation, move the entry to the Fixed topic. Delete any workaround that was documented with limitation.

Fixed Limitations Criteria

Before you write a fixed limitation, verify that it meets the criteria for the release notes.

Most of the time, bugs are fixed in the code, tested, and closed. However, Dev might close a bug for multiple reasons. For example, it might be a duplicate of another bug, or it might be closed with a workaround. When you review a closed bug, you need to read the closing comments at the end of the notes to find out why a bug was actually closed. If it is not checked in to the code, we cannot publish it as being closed.

When you review fixed limitations that are release note candidates, consider the following criteria:

Fixed release

Verify that the limitation was verified and closed in the release that you are documenting.

Note: If the limitation was linked from a previous release and this fix was merged from a previous release, do no document it again as fixed. The content reference informing customers about fixes in previous releases is sufficient.

Status

Verify that the status of the limitation is "closed." If it is "resolved," verify with Dev and QA that they will be able to close it for the release. Watch the bug for the status change.

Content to include

Include fixed limitations that meet the following conditions:

- The fix was checked in to the code.
- The bug was not opened in the current release.
- The bug was previously documented as a known limitation, or it was reported by a customer or GCS.

Content to exclude

Do not include fixed limitations that meet the following conditions:

- If a limitation is closed as a duplicate, look at the linked bug to see if it belongs with known limitations.
- If a limitation is closed as "will not fix" or "as-designed," consider documenting the issue in the Knowledge Base or user documentation. If the behavior still seems buggy, put this in a Knowledge Base article. If the workaround, or the user actions required to get the desired behavior are fairly simple, consider including it in the user docs.
- If a limitation is closed with a workaround, you can include it in a Knowledge Base article. Workarounds can include undocumented flags, registry edits, and system patches.
- If an issue was issue that is closed with a custom property, it is a candidate for the Knowledge Base. We do not expose these properties to all customers, so any documentation will be through an internal KB article. Ask Global Customer Support and QA if they want this documented.
- If a limitation is closed with a note that Documentation is adding it as a limitation in the release notes, let Development know that we cannot document anything in the release notes without Dev commitment to fix.

Rules and Guidelines for New Features

When you document a new feature in a release guide, ensure that the content follows the standard structure.

Rules and guidelines for all release guides

Consider the following rules and guidelines:

- Use the following lead-in sentence for each version topic: "This section describes new features in version <*version*>. "
- Alphabetize the sub-topics and sections in each functional category topic, unless you receive different guidance from product management. PM might occasionally want to highlight certain information first.
- In general, do not document new UI features. Document changed behavior, not changed appearance. For example, if there is a new dialog box for unlocking repository objects in Developer tool, add an entry about unlocking objects in the Developer tool. The exception is if the UI for the entire tool has changed.
- Optionally, add images to show UI or other changes when you think these will benefit the user.
- Include a book reference for each feature.

Use the following syntax for the book reference: "For more information, see the <*full book name, including version number*>." Use the <cite> tag for the product name, version, number, and book name. For example, "See the *Informatica 9.6.1 Administrator Guide*.

- Optionally, include a chapter reference for a feature.