B Center on
Health Policy
at BROOKINGS

February 23, 2026

Frank J. Bisignano
Chief Executive Officer, Internal Revenue Service
Department of the Treasury

Daniel Aronowitz
Assistant Secretary, Employee Benefits and Security Administration
Department of Labor

Robert F. Kennedy, Jr.
Secretary
Department of Health and Human Services

Re: Transparency in Coverage [CMS–9882–P]

Dear Mr. Bisignano, Assistant Secretary Aronowitz, and Secretary Kennedy:

Thank you for the opportunity to comment on your agencies' proposed revisions to the Transparency in Coverage (TiC) regulations.[1] Our letter argues that adopting an appropriate standardized file format, such as Apache Parquet, is the single biggest step that the agencies could take to make the TiC data more usable. The key advantage of a standardized file format is that it would relieve data users of the need to write custom code for each payer's files.

However, the choice of standardized format matters greatly. The JSON format that most insurers currently use is cumbersome for data users because it stores data inefficiently and is computationally costly to parse. CSV, the other format that the agencies discuss in the proposed rule, would likely outperform JSON, especially if the TiC schema's nested structure were replaced with a multi-table relational structure. However, Apache Parquet would likely be a much better choice than either CSV or JSON. Parquet is explicitly designed to facilitate efficient reading, writing, and storage of large datasets, features that would allow for much lower file sizes and processing times. Parquet can also represent nested data structures (although it would likely still be advisable to move the TiC schema partway toward a relational structure alongside the change in file format). Importantly, Parquet also enjoys broad programming language support.

We also offer comments on two of the other proposals in the proposed rule:

- Including enrollment and utilization information in the TiC data could facilitate various types of analyses that are currently impossible. However, maximizing the utility of these data would require some modifications to the agencies' proposals.

[1] The views expressed in this letter are our own and do not necessarily reflect the views of the Brookings Institution or anyone affiliated with the Brookings Institution other than ourselves. We thank Loren Adler for comments on a draft of this letter, Ben Graham for research assistance, and Rasa Siniakovas for editorial assistance.

- Excluding service codes from the TiC data based on providers' taxonomy codes could reduce file sizes and increase usability, but it is unclear how much. There are also steps that the agencies could take to mitigate the risk of inappropriately excluding service codes.

The remainder of this letter examines these points in greater detail.

Adopting a standard TiC file format
The proposed rule seeks comment on whether it should require all TiC files to be published in a single standardized format and, if so, whether that format should be JSON or CSV. There is, in our view, a strong rationale for adopting a standardized format, as this would relieve data users of the need to write custom code to handle each payer's files. But the choice of format matters greatly.

In what follows, we argue that JSON is a very poor choice for the standardized format. While an appropriately crafted CSV format could outperform JSON, there are other formats—such as Apache Parquet—that could substantially outperform both JSON and CSV.

<u>Shortcomings of JSON</u>
The agencies state that they seek a file format that performs well at each stage of the extract, transform, and load process. In our experience, JSON is an exceptionally poor performer at the "load" stage of the process. There are, in our view, two fundamental reasons for this.

First, JSON encodes information in a very space-inefficient way. Field names are repeated every time a field appears, and the file's hierarchical structure is encoded using delimiters that are repeated on every individual record. In one In-Network File that we recently processed, which we believe is typical of JSON TiC files, these structural elements alone accounted for around three-quarters of the file's characters. JSON also encodes all information as plain text, which for many types of data (e.g., numerical data) is much less efficient than using an appropriate binary format.

These inefficiencies in how JSON encodes information are a major reason that TiC files published in JSON format are so large. And as correctly noted in the proposed rule, large file sizes create major challenges for data users. In particular, this means that these files are generally too large to be held in a typical computer's memory. As a result, ingesting TiC files frequently requires specialized hardware, specialized programming knowledge, or some combination of the two.

These large file sizes also make the agencies' statement that JSON "enjoys native support" in most programming languages misleading as applied to the TiC files. While most languages do include libraries for working with JSON files, these libraries often require that the entire file be loaded into memory for parsing. As noted above, this is often infeasible, at least for users without specialized computing facilities. Indeed, when working with TiC files, we have typically had to rely on libraries that can parse files on a "streaming" basis (that is, a little bit at a time). These libraries are not as widely available and, in our experience, more cumbersome to use.

Second, because JSON encodes the file structure in plain text alongside the actual data, that structure must be reconstructed as the file is loaded by parsing the field names and delimiters. Given the large size of the TiC files, that parsing step adds substantial computational burden.

Alternatives to JSON

There are alternative file formats that would substantially outperform JSON.

CSV format

One potential alternative file format is discussed in the proposed rule: CSV. Because CSV avoids repeating field names every time a data element appears, a CSV format has the potential to result in files that are much smaller and, in turn, easier to work with.

Importantly, achieving good results with a CSV format would require thoughtful implementation. As the agencies note, a downside of "flat" tabular formats like CSV is that they cannot directly represent "nested" data structures (like the current TiC schema) where "higher" nesting levels contain information that applies to multiple records at "lower" nesting levels. As a result, naive approaches to representing TiC data in CSV format would repeat information that appears at higher nesting levels on every record below it in the hierarchy, which is inefficient.

However, this problem can be avoided by adopting a multi-table relational structure in which information that appears at different nesting levels is stored in distinct tables that can then be linked together. This largely avoids the repetition problem because all that needs to be repeated in the tables that correspond to the lower nesting level is a pointer to the appropriate row of the table for the higher nesting level, which can typically be stored quite efficiently.

Apache Parquet format

While an appropriately crafted CSV format would be a substantial improvement over JSON, there are alternative formats that would likely perform even better and still offer broad software support. We view Apache Parquet as one especially good option, although there are likely other formats that could also perform well in this application (e.g., Apache Avro or HDF5).

Parquet is explicitly designed to make reading, writing, and storing large datasets highly efficient, including by encoding data in efficient binary formats and incorporating high-performing compression algorithms. Parquet can also directly represent nested data structures like the TiC schema without the need to convert to a multi-table relational structure. And Parquet enjoys very broad programming language support, notably including Python, R, Java, and C++.[2]

While Parquet is fully capable of representing nested data structures, there would likely be advantages of moving the TiC schema partway toward a multi-table relational structure in conjunction with adopting Parquet. Consider, in particular, the In-Network File. The current TiC schema for that file specifies a top-level object that contains several metadata fields, an array of objects representing negotiated rates, and an array of objects representing provider identities. Placing the two arrays in separate tabular Parquet files and storing the metadata inside each file using Parquet's metadata functionality would result in Parquet files that are far easier to read and

[2] For more information on libraries providing language support for Parquet, see Apache Software Foundation, "Sub-Projects," January 17, 2025, https://parquet.apache.org/docs/contribution-guidelines/sub-projects/.

write because the In-Network File would no longer consist of a single very large data row (which is typically less efficient to work with than multiple smaller rows containing the same data).

We have experimented with translating TiC In-Network Files that are currently in JSON format into Parquet files that use the modified schema described above. We found that this approach greatly reduces both file sizes and processing times. For example, we recently processed a TiC file in JSON format from a major insurer that was 1.4 GB when compressed and 31.9 GB when uncompressed. By contrast, when storing the same data in Parquet format, the files had a combined size of 1.1 GB. Read times were also much faster. Whereas parsing the compressed JSON file into a form suitable for analysis took multiple hours, loading the Parquet file took seconds.

Reporting enrollment and utilization data

The agencies propose to require insurers to report plan enrollment and utilization data alongside the TiC data. We believe that data like these could be quite useful to data users, but, in each case, some changes to the agencies' proposals would be necessary to realize this potential.

<u>Enrollment data</u>

As the agencies correctly note in the proposed rule, the fact that insurers do not currently report plan enrollment as part of the TiC data makes these data less useful than they would otherwise be. Notably, it makes it difficult to compute meaningful market-wide averages because it is unclear what weight to assign to different plans' prices. More generally, it makes it difficult to estimate how many people are enrolled in plans with particular features or how plan features vary with plan size. Requiring insurers to report enrollment data could help facilitate analyses like these.

However, the agencies' proposal that enrollment data be reported at the network level on the In-Network File would limit the usefulness of these data. In our experience, many plans have multiple In-Network Files. Where this is the case, plan enrollees would end up being counted multiple times (once for each In-Network File), which would make these enrollment data harder to interpret and harder to use. A simple alternative that would avoid this problem would be to require insurers to report enrollment data at the plan level on the Table of Contents File. This should not place meaningfully greater burden on insurers, as insurers would presumably need to tally enrollment at the plan level in order to generate the network-level tallies envisioned in the agencies' proposal.

<u>Utilization data</u>

Including utilization data on the TiC files could also make them more useful. As the agencies note in the proposed rule, these data would permit data users to identify which providers are routinely seeing a plan's enrollees. This may be of direct interest to many data users, especially for those interested in understanding the degree to which different plans offer access to services. Like enrollment data, utilization data would also ease the process of computing meaningful market-wide average prices by making it clearer what weight to assign to each negotiated price.

However, the agencies' proposal has one important limitation. Specifically, insurers would be required to report only a binary indicator for whether a particular provider delivered a service via the provider network in question at least once in the prior 12 months, rather than the number of times that service was delivered. Particularly for provider networks that serve a large number of enrollees, a provider may often deliver a service at least once, even in instances where a provider does not routinely serve a plan’s enrollees. For this reason, a binary indicator like the one the agencies envision would not be especially informative and could serve the purposes outlined above to only a very limited degree. Actual utilization tallies would be far superior. If the agencies are concerned about small cell sizes for privacy or other reasons, they could suppress these data (e.g., by simply indicating that the service was delivered at least once, but less than 11 times), an approach similar to the one that they propose to take for out-of-network data.

Excluding service codes based on provider taxonomy codes

The agencies propose to require insurers to exclude services that a provider is unlikely to deliver from the In-Network File. Under the proposal, insurers would use a provider’s National Uniform Claim Committee (NUCC) taxonomy code, as well as their internal guidelines for which types of providers are eligible to be paid for a service, to determine which provider-service pairs to exclude.

Whether this approach will achieve the agencies’ goals of reducing file size and, in turn, improving TiC files’ usability hinges on how permissive insurers’ mappings from taxonomy codes to service codes typically are. There are plausible reasons for insurers to make these mappings relatively permissive. For example, for taxonomy codes corresponding to multi-specialty group practices, virtually all service codes would likely need to be included to accommodate the wide range of services potentially delivered by these types of entities. And insurers may choose to use permissive mappings in other settings as well to accommodate errors in providers’ taxonomy codes or instances where providers deliver care that straddles multiple specialties. If these mappings are, in fact, highly permissive, then this approach might achieve only modest reductions in file size, in which case it might not be worth implementing. Thus, information on what insurers’ mappings typically look like would be valuable as agencies decide whether to finalize this proposal.

If the agencies do opt to move ahead, another risk (which is, in some sense, the opposite of the risk outlined above) is that this approach may sometimes exclude some provider-service pairs that are actually relevant to data users. There are a couple of straightforward steps that the agencies could take to mitigate this concern. First, the agencies could require that the In-Network File include any provider-service pair with positive utilization during some historical period, even if that pair would otherwise be excluded. This could capture instances where insurers have sometimes overridden their general taxonomy-code-based code mapping during the claims adjudication process. Second, the agencies could require insurers to report the taxonomy code they have assigned to each provider for filtering purposes. While this would not directly mitigate the loss of relevant data, it could help data users identify and handle potential data errors.

Thank you for the opportunity to comment on this proposed rule. We hope that this information is helpful to you. If we can provide any additional information, we would be happy to do so.

Sincerely,

Matthew Fiedler
Joseph A. Pechman Senior Fellow in Economic Studies
Center on Health Policy
Economic Studies Program
The Brookings Institution

Yihan Shi
Research Analyst
Center on Health Policy
Economic Studies Program
The Brookings Institution