

### Folder and file naming best practices are essential

The folder and file name combination is the most fundamental method for searching, identifying, and retrieving electronic documents. It is analogous to using a person's name and address to help locate and identify an individual. This is not an earth shatteringly new concept but knowledge-workers commonly overlook the strategic importance of proper and consistent folder / file naming to [manage documents](#) more diligently. The message is simple: "To easily find electronic files, just ensure that all stakeholders name and store files properly and consistently". If you believe that everyone is already doing a great job in your organization, they can do even better by using a powerful yet simple technique – "packed-element" file naming.

### Folder and file naming using packed-elements

For a quick and general overview of structured folder / file naming and best practices you may refer to a previous article: [10 rules for Folder and File Naming Best Practices](#) available on the [www.exadox.com](http://www.exadox.com) website. To take your folder/file naming best practice to the next level, you should consider taking strategic advantage of "packed-element" file naming in your folder / file naming policy. Packed-elements (*my own term*) are structured information-dense folder / file naming elements that are created by consolidating multiple classification components (codes) into one single element, where each classification component has an established format and is taken from a designated repertoire or list. E.g. **SMIJ094HR** (Format: AAAANNNA) or **SMIJ-094-HR** can be a subfolder or file name element encoded for John **SM**ith employee no. **094** that works in the **H**uman **R**esources department. Basically, one could think of a "packed-element" as a folder or file name element on steroids because it can be jam packed with a lot of info-muscle. A single packed-element would be more than ample for a folder name; however a file name may include several elements allowing the possibility to have more than one packed-element. The end-result is an information-rich folder or file name that could be exploited for fast and precise file searching, identification and retrieval.

### More examples of folder / file naming using packed-elements

As an example, let's assume that you wanted to design the encoding of a packed-element to help you better classify and identify documents relating to a specific individual. You would want the folder or file name packed-element to include their name as primary classification component, followed by the city where they live, and finally their date of birth. So, your packed-element components for John Taronosky, that lives in Detroit, and born on June 29<sup>th</sup>, 1975, could be encoded as follows: **TARJ-DE-19750629** or **TARJDE19750629**. This particular example was chosen for illustration purposes because it is easy to relate to, but at first glance it could appear overwhelming. The resulting packed-element does not have to be overwhelming. Let's look at a simpler example: A facilities manager that manages projects for multiple buildings in several American states could design a packed-element for his **FacilityID** element that only consists of 2 classification components. The first is based on State and the second on the building number. E.g. **CA-06** or **CA06**, where California is the state and 06 is the building number 6. Alternatively, if they must include the city because they have several buildings in each city, it could be encoded as: **CA-LA-06** or **CALA06** where **LA** is for Los Angeles.

Good file name structures always consist of several elements and it is possible to include more than one packed-element in a given file naming convention or policy. If we continue with the hypothetical facilities management example, the full file name can be structured as follows:

**FacilityID\_Project ID\_StartDate\_Activity\_DocType\_FreeText.ext** where both the **FacilityID** and **ProjectID** can be packed-elements. We already saw what the **FacilityID** can look like. This is what the **ProjectID** packed-element can look like: **109OF12REN** or for short **109O12R** where this denotes that the assigned project number is 109 and this is for the renovation of the offices on the 12th floor. Note: If the renovation is for the basement, **BASE** can be substituted for **OF12** or **BAS** for **O12** to maintain the same number of characters. In the file name structure above, **Activity** may not require a packed-element because it may be a single task such as **PLA** for planning phase, **DES** for design phase, **CON** for construction phase, etc... However, if there is a need to also include sub-activities such **EL**-electrical work, **PL**-plumbing or **HV**-heating and ventilation, the **Activity** element could also be a packed-element that would look like **DE-EL** denoting the activity is relating to the design phase of the electrical layout. The end result is an information-rich folder or file name thanks to information-dense packed-elements.

### Structuring packed-elements for folder and file naming -5 rules

The following are some general rules when structuring the encoding of packed-elements:

1. Limit the number of types of classifications components to only 2 to 4 max, if you need to absolutely go beyond 4, consider adding an additional element or including a major separator. **Do:** CALA06\_109OF12REN. **Don't:** CALA06109OF12REN. **Reason:** *Less is more.* The beauty of using this technique is that you can instantly recognize the meaning at a glance when looking at the contents of the packed element. If you have to struggle to decode it due to information overflow, the benefit can be lost.
2. Just as in file naming, start the classification components from most significant to least significant. **Do:** CA-LA-06. **Don't:** 06-LA-CA. **Reason:** *It is logical.* Anytime you want to zero-in on something, you always start from the broader information and work yourself toward the finer details. It also helps when sorting.
3. Use the least number of characters for each classification component. **Do:** CA-06. **Don't:** CALIFOR-BLDG06. **Reason:** *A little may be enough.* The initial instinct is to use large or full descriptive codes that are almost idiot-proof. This is a safe approach if you are targeting a general community; but it may not be necessary and certainly overkill for initiated stakeholders.
4. Use hyphens between classification components if and when absolutely necessary and avoid using the same separator as being used between elements. **Do:** CA-LA-06\_109-O12R. **Don't:** CA\_LA\_06\_109\_O12R. **Reason:** *Give them a break.* The eyes and mind need to compartmentalize as much as possible. By not facilitating that with the benefit of a differentiating separator it could be overtaxing and not appreciated.
5. Keep the number of characters for each classification component consistent and small. **Do:** NY-NYC-10 and MI-DET-06. **Don't:** NY-NYC-10 and MICH-DETR-6 (inconsistent). **Reason:** *Stick to it.* Once you have decided on the number and type of characters required for a given classification component, for conformity sake and to help maintain sanity, it is best to adhere to an established coding format. Sorting is also affected by inconsistencies in the structure.

### Packed-elements benefits

At this point you have a general idea of what packed-elements look like. Here are some of their major benefits:

1. Information – lots of detailed information within a single element.
2. Standardization -everyone will be using identical pre-authorized packed-element identifiers.
3. Concision – the efficient compression of multiple classification components.
4. Structure – encoding of packed-elements is as per a prescribed format.
5. Control – packed-element list creation and management can be delegated to key personnel.

**NB:** These benefits can only be realized if everyone consistently adheres to the same folder and file naming policy and they populate folder and file names with the proper data. Therefore, folder and file naming training and monitoring are implicit necessities, especially if folder and file naming is being done manually. A safer and more efficient alternative to a manual approach is to take advantage of semi-automated folder and file naming. For information on this concept you may refer to the White Paper: [Semi-Automated Structured Folder and File Naming](#).

### Conclusion

Instituting or improving on folder and file naming best practices can be extremely rewarding in terms of productivity improvement, nevertheless, there are always challenges that must be surmounted especially when multiple stakeholders are involved. One of these challenges is that it will take time and effort to introduce and plan your new folder and file naming policy. Once your policy is in place, the next major challenge is ensuring that everyone follows the naming policy diligently; the best way of achieving this is by taking advantage semi-automated file-naming tools such as [eXadox DT](#). To help you with your policy planning effort you may download a 10-day trial copy of [eXadox WIZ Filing Policy Planning Wizard](#). It will expedite your planning process, and it is free.

**About the author:** Vincent Santaguida is the CEO and founder of MultiCIM Technologies Inc. He was the driving force and principal architect for [eXadox™](#), a software tool that helps name, organize and manage electronic documents and computer files. He writes [articles](#) and speaks on subjects dealing with paperless office and information asset management strategies as a means of encouraging productivity.

[www.exadox.com](http://www.exadox.com)