

# **IP packaging based on XML signatures**

**G. Saucier, L.Ghanmi, M. Hamdoun**

**Design and Reuse**

Didier Maurer

**Dolphin integration**

**Grenoble, FRANCE**

## **Abstract:**

This paper presents “best practice” up to a “standard” for packaging and delivering IP versions/configurations based on “XML signatures”; A XML layer is created on top of the files defining virtual IP version/configurations. When delivering an IP to a consumer he gets in addition to the files its “XML signature” providing him a comprehensive view of the delivered files. It is shown that this XML signature or stamping process prevents from redundant storage at the supplier site, provides the bulk of a very comprehensive; unified, modular IP management environment allowing for instance “automated” shipment to customers, non ambiguous reception of an IP version by a consumer as well as outstanding consumer support features.

## **Introduction:**

Two features are commonly discussed when addressing IP Packaging and Delivery namely the completeness in terms of delivered files and the quality of the delivery. In others words, “What should be delivered in terms of set of files?” and “How good is what is delivered?”

This paper addresses a complementary facet namely: “How an IP version/configuration should be delivered” in order to make its delivery straightforward, facilitate its integration by the consumer and provide reliable and efficient support and update;

The problem is known to be complex for 4 main reasons:

- First, an IP is the result of a design process handled by a design data management system (DDM); An IP is a complex electronic subsystem. Its delivery, as a reusable package, contains a large number of files, corresponding to views at various levels of abstraction. It requires identifiers in terms of version/revision often refined at the package levels.
- Second, the IPs have to be hardened on the technology/process/library requested by the client and this may lead to a large number of delivered versions
- Third, most of the IPs are described as generic and configurable objects and giving values to parameters or selecting configuration options creates “IP instances”. Therefore distinct instances or configurations will be delivered to various IP consumers. A large number of architectural configurations for instance may exist for the same generic IP.

- Last, the provider may have to tune the IP delivery to the consumer environment namely RTL language, formats, EDA tools, simulation environment...

It is therefore easy to understand that managing version/revision/configuration/environment options needs adequate management features.

An optimized delivery process is the key for both the provider and the consumer as time to market pressure for an ASIC or SoC products is high and this requires high quality delivery mechanisms pressure for an ASIC or SoC products is high and this requires high quality delivery mechanisms

\*At the consumer site, fast, reliable and automated reception of a clearly identified understandable set of files has to be provided

\*At the provider site, IP delivery and support to customers have to be mastered with advanced techniques to reach the expected level of customer satisfaction. In simple words the “right” configuration has to be defined with appropriate version control to a customer followed by an adequate support. If a bug occurs, the “right” consumer should receive appropriate warning and incremental delta file set if relevant

In conclusion, describing, defining, packaging and delivering the various options of an IP can raise quality issue in the delivery and support process to consumers.

To solve such a problem, this paper proposes a mechanism for identifying IP version/ configurations through “XML signatures”, preventing from redundant storage and providing straightforward IP packaging, delivery as well as supporting all related IP management functions. It relies on the creation of a XML layer between the database managed by a DDM tool and the IP packaging and delivery management environment. This layer should hopefully to solve all the major issues.

The whole process is illustrated on a real case namely the delivery of a configurable processor of Dolphin Integration; the Typhoon 80251 processor

The proposed IP packaging and e\_Delivery mechanisms described in this paper have been implemented using industrial tools from D&R (IP Provider Station<sup>TM</sup>, IP Consumer Station<sup>TM</sup> and IP Reuse Station<sup>TM</sup>) These tools offers a broad set of about 50 modular IP management functions providing fast, reliable and hopefully a standardized process IP packaging and delivery management.

## **Section 1 Basic mechanisms for IP packaging and delivery**

### **1. Definitions**

#### ***1.1 Static e\_Version/Revision and client configuration of an IP***

IPs similarly to any design is the result of a complex design flow process put under revision control; Such a design process commonly uses a design data base management (DDM) system.

Once an IP version/revision has been released in the native database, it comprises a large number of files including files related to some intermediate design or validation steps. Only a subset of these files constitutes the reusable IP package and will be sent out to IP consumers.

Such a set of files, dedicated to be sent out as deliverables over a network (Internet/Intranet), to a consumer or another designer in a collaborative design environment, is called in this paper an **e\_Version**, the “e” prefix emphasizes the “exchangeable” feature over the net through online download or ftp mechanism.

Thus, an IP e\_Version /Revision is defined by a subset of files of a native released IP version in the design database, each package or file is identified by its paths or tag bridging of the native directory.

An IP version can be a soft IP containing only VHDL views or refined to a technology/process library and therefore will be labelled with the technology/process/library information. It can also be related to a specific EDA / format environment.

From an IP version/Revision called “reference version”, several subsets of delivered packages can be extracted in terms of files content linked for instance to a business agreement. As an example, an IP can be delivered both through its VHDL model only for simulation but also through a VHDL model together with as a netlist (firm IP) for integrating in the design.

Specific consumers may ask for simulation or validation results performed with specific tools and some others may “even” ask to get the synthesis scripts. Note that some packages (Core model, for instance) belong to all delivery while optional files/packages are identified through configuration options.

## 1.2 Configurable IPs

A large number of IPs are in fact created by a generator. At a high level of abstraction, this may correspond to architectural options. These options are diversified and application specific; Let’s mention memory architecture option, processor option, polynomial selection for encoder/decoders etc. It is of course interesting to satisfy a large number of customers in terms of architectural options often related to cost/performance trade off.

As an example, the configurable 16-bit processor from Dolphin Integration (80251-Typhoon), which is a “Soft IP”, can be delivered with a large numbers of options (each peripheral can be selectable individually and some processor’s option, like the number of interrupt, can be configured). About 1.327.104 configurations can be selected and delivered to customers. This means that the IP is in fact a parameterized VHDL model generator and the set of parameter values are entries of the generator.

## 2. XML Signature of a static e\_Version/Revision/Configuration

### 2.1 XML signature

An XML signature of an IP e\_Version (Revision) is a XML file organized around a XML DTD (Document Type Definition) representing the IP directory hierarchy the leaves of which are:

- XML labels whose values are paths to the native files /packages
- Attributes characterizing the version/revision
- Optional Attributes for supporting Management primitives

In some environments, the XML DTD of the signature can be automatically generated from the DDM environment as a XML file.

Each file /package is “XML stamped” by 2 static identifiers.

- A (global) IP version/Revision id, which is the IP version id
- A (local) version id imported (tag or header) from the design data management and allowing to share later on \* a package between versions

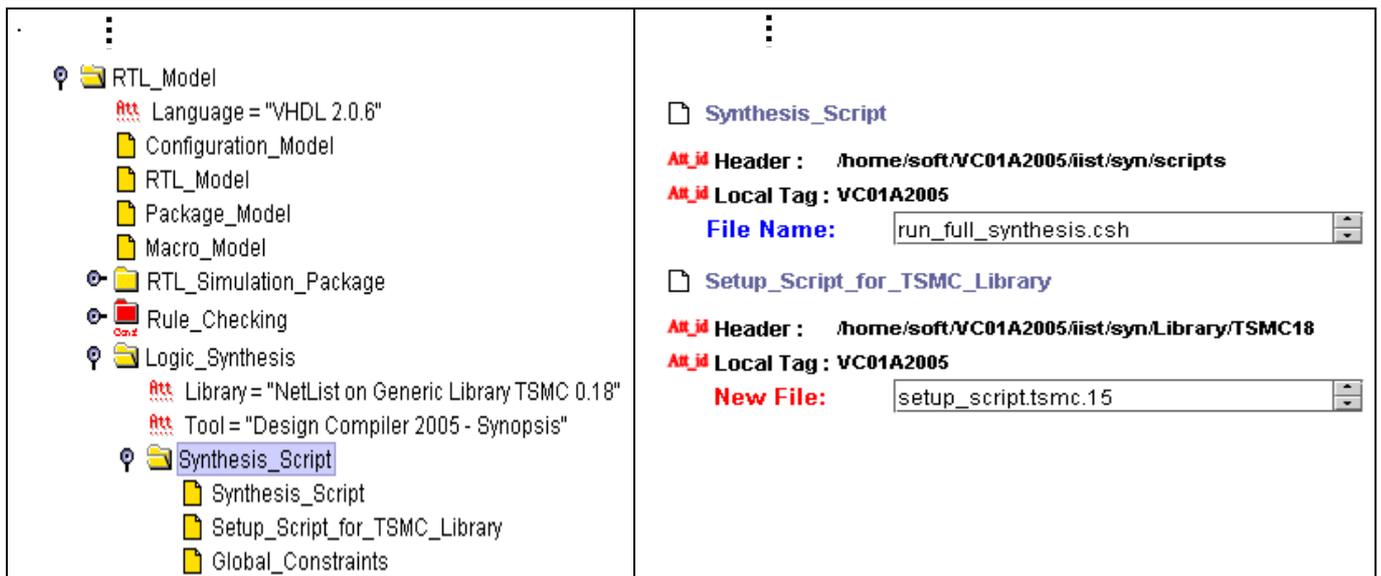


Figure 1: XML signature of an IP version

## 2.2 Delta between 2 versions and non-redundant storage

XML signatures are associated similarly to incremental directory content (Delta) between 2 versions/revisions

The underlying idea of this approach is that IP Versions should be defined by their signatures and not by the actual files and that redundant storage is avoided.

Suppose that 2 IP version share the same VHDL package, then the global version id for the VHDL package will be different in the 2 IP versions inheriting the IP version id whereas the local identifier (pointer to the DDM) will be identical; thus in no way the VHDL will be stored twice.

In figure 1 on the left side a small portion of the XML signature tree is displayed. It should be understood that the considered e\_Version contains the RTL model and simulation as well as the synthesis script on a generic TSMC library; In red, it should be understood that in this precise e\_Version the rule checking package is not part of the delivery; the attributes give all relevant information about the used environment

On the right side the file, names are indicated as well as headers of the files and tags possibly reporting to the native DDM database.

## 2.3 Static Configuration of an IP version

The XML signature of a static configuration is a subset of the directory tree of the IP version. The configuration is identified by a set of symbols; each symbol being associated to an optional package belonging to the configuration

For delivering an IP configuration, an extraction/compression step will automatically extract and compress the files belonging to configuration into the delivery platform.

After delivery, the delivery files can be removed from the server to prevent from overloading the server.

In conclusion the consumer configuration does not need to be stored as such; only the signature has to be stored for managing later support.

The management of the packaging and delivery is illustrated in figure .3

Red edges represent actual files transfer whereas yellow edges show the XML signatures creation and transfer. In the origin data base, the full snapshot of the design flow release is identified and XML tagged, the first layers point to the files of the e\_version which are then filtered by the configuration signature, the resulting files are extracted and stored in the delivery server for download or, shipment.

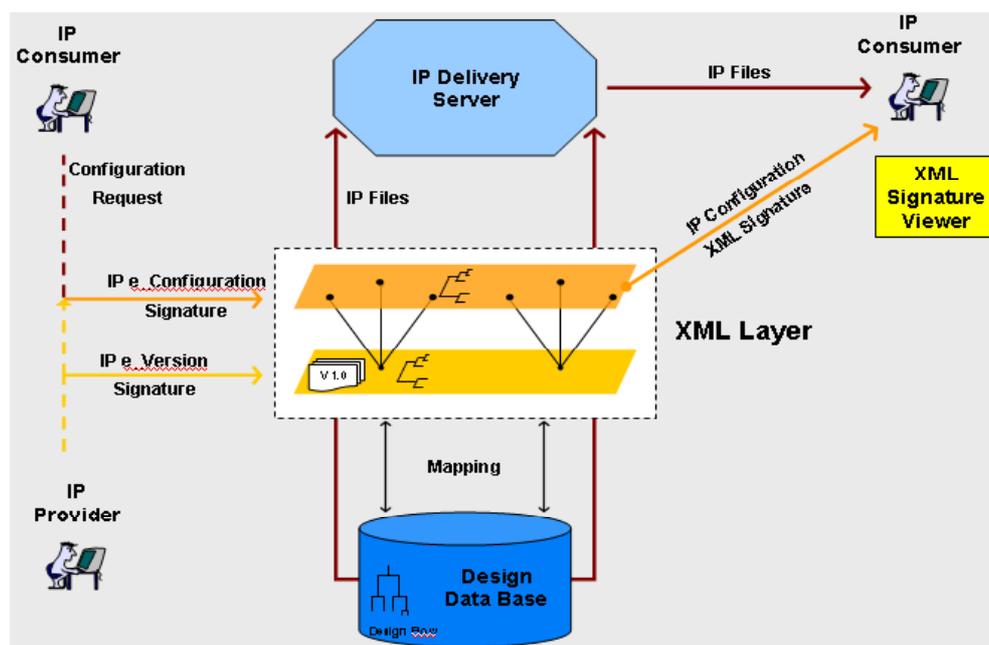


Figure 2: IP file transfer and XML signatures of a static e IP version

### 3. XML signature and Configurable IP

A generator as mentioned above has commonly a high level entry where allowing the consumer defines the configuration he wishes to get;

The configuration options are then translated automatically into entries of the generator as well as to a XML signature defining the configuration in terms of packages/files.

The generator will be triggered and generates the output files which like previously will be compressed and put on a delivery server for actual delivery through download, shipment or ftp; A shipment is identified by the client order. To facilitate the support an automated back annotation process will insert a tag which is the order id in the files in the DDM that are part of the delivery and shipped to the customers; Thus at any time the set of delivered files are identified without having to store them explicitly.

Example:

In this example of figure 3, the XML signature of a configuration of the version “*Flip80251TyphoonRev 1.3a*” is illustrated; the order of the configuration is represented at the right side and translated in a symbol of strings

On the signature of the left side, the yellow packages are the packages shared by all configurations (obligatory files); an optional package is identified by a symbol; the symbol A corresponds for instance to the “standard peripheral with at least one PCA module“. Once selected by the user, they appear in red.

The image shows a configuration tool interface. On the left is a file tree for an IP profile. On the right is a configuration panel with three sections: Configurable Features, Controller Peripherals, and Add-ons. Each section contains various options with dropdown menus or checkboxes.

Section	Option	Value
Configurable Features	DMA Support Interface	Yes
	Protection Of Access To Memories PAM	No
	Additional Interrupts 7 External IT	No
Controller Peripherals	Timers	0
	Watchdog Timer	without
	PCA Modules	5
	UART	without
	IO Ports	4
	Port Direction Signals	without
	ECPMU	with
Add-ons	I2C	Without
	SPI	without
	HDLC-FIFO	without

In the example below the user asked for a standard peripheral but did not ask for an UART and consequently in the XML signature the package appears in blue.

An order is defined by a string of symbols identifying the optional packages to be included.

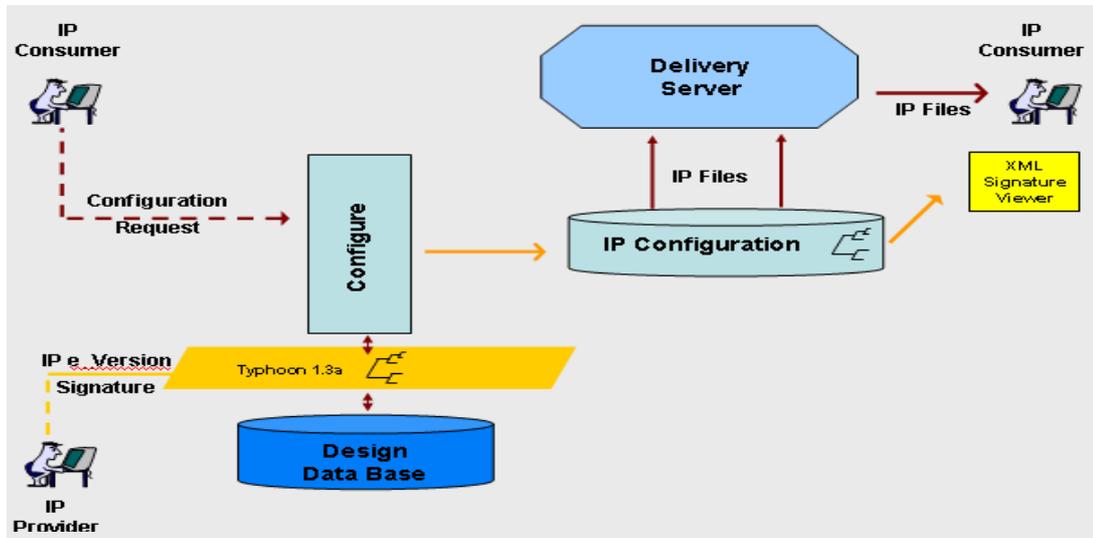


Figure 4: An order automatically processed and memorized

The figure 4 illustrates the packaging and delivery process ;In this case the XML layers does not operate as filter for selecting files but triggers the generator and back annotates the files with the customer id for further management.

#### 4. XML signature based management features

The benefits are tremendous and D&R is delivering about 50 modular management functions written in Java and operating on such XML signature layers which can be used independently.

##### 4.1 Non-redundant File storage

First of all, it should be recalled that as explained above, the XML signatures are in fact stored on an upper management layer on top of the IP database. They identify through a bridging mechanism, virtual IP version/revision/configuration:

Two version sharing file /packages exhibit identical local identifiers for the shared packages and configuration delivered to customers do not need to be stored ;Once shipped only their signatures is required to store the knowledge and provide a support basis; in other words the signature contains enough information to restore the data of the configuration.

##### 4.2 IP packaging and delivery control

The benefits is directly inherited by a good identification of the object (through its **signature**)

- Automated file packaging: file extraction compression, transfer to a delivery station followed by automated delivery
- Back annotation in the design data base with an order/client id
- Translation of procurement conditions (contract, corporate agreement, contract content) into a configuration generation and assignment of Download rights
- Download/Shipment panel with a view of the files sent to a consumer

### 4.3 *IP version visibility*

When receiving an IP version the consumer has access to the signature (directory tree with all the flow attributes. This will save significant time in receiving the files and integrating the IP in his own design environment).

Mirror catalog can also be created at the customer site;

### 4.4 *Support and Bug management*

- Flexible bug formatting through extended XML formatting
- On line shared bug management data base
- At any time the delivered files sent to a customer are well identified; once a bug is located and associated to a file, an automated process can find all versions /configuration affected as well as all consumers to be warned.

## 5. **Conclusion and standard proposal**

It should be first noticed that the features described below constitute a comprehensive collaborative design environment and IP based SoC project management is easily added on top.

- Should there be any standardization of the delivery process be proposed?

↳ The notion of packaging and delivery standards can be understood at several levels:

- Loose standardization

It could be stated that an IP version/configuration cannot be delivered without its attached XML signature and that D&R promotes its delivery as a standard for IP delivery.

- Medium level of standardization

XML labels are proposed for standardization and a minimal set of characterization attributes identified.

- Strong standardization

The next level of standardization considers imposing a structure or content of the delivery; this means that the signature is a XML scheme or XML DTD (Document Type Definition) imposed by a body or a corporation or by the large company acquiring this IP. This requires a mapping from the native environment to the reference signature scheme or DTD.

A specific function is included in D&R tools performing a mapping between the native signature labels and structure and the standardized signature labels: In other words, All delivered IPs will use the same XML labels, or same DTD or scheme and same attributes.

In conclusion, XML catalogues supporting massive data import from database has already become a reality the last decade.

By “standardizing” XML signature based IP delivery for delivering IP files will become promptly the next standard or best practice.