MaRS, ACES and IPArch Container Setup

Abstract

A Guide to Setting up and Installing MaRS, ACES and the IPArch Container with ProMark

Nick Leuci

Intel Corporation

March 2023

Contents

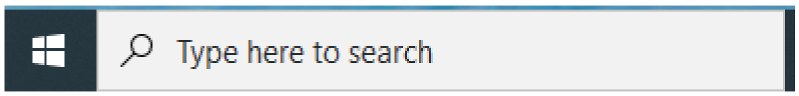
1. Prerequisites

MaRS is an Intel initiative to rigorously automate the entire IP design process from specification to fabrication. The core concept is building a system for **Machine Readable Specs**, or **MaRS**. This document is not intended as a reference or even an introduction to MaRS itself, though a list of relevant reference links will be provided.

This document focuses on installing and setting up the ***entire*** set of tools and frameworks necessary to work with MaRS, ACES and the IPARCH container. There are several Intel websites with procedures to do these various installs. This document is supplementary to them, but it will refer to them extensively. This document's main use is providing a comprehensive view of setting up and configuring MaRS, ACES and the IPARCH container for those getting started in this engineering arena.

ASSUMPTIONS
There are many ways to use the ProMark **MaRS** framework at Intel. This document assumes you are using a Windows PC connected to the Intel VPN. You will also need to have proper authorizations and permissions to access some of the sites and tools cited. The Intel AGS provides the means to get any needed authorizations. Use the Intel 1Source site to get a summary of your relevant authorizations.

All the tools cited require installations. Some of these tools will have download or installation links for you to access. Other tools will require you to go to the Intel Software Market to find and download the tool's setup. Note that the Intel Software Market is gradually being phased out and replaced by the Intel Company Portal. You will currently find both facilities by ***left-clicking the Windows Start Menu*** on your PC, in the lower left corner of your screen. The Windows Search Control is immediately adjacent to the right of it, as shown below. You can use the search control to find already installed features and applications.



Each of the components described in this document have specific requirements and authorizations. Here is a general annotated list to introduce you to most of them.

- **ProMark.** ProMark is Intel's proprietary internal web publishing technology using an open-source Markdown language framework. It is based on a collection of Python-developed automation tools that allow architects to write specs as text files in the Markdown language, use a Git repo for revision control and collaboration, and publish final specs as either HTML or PDF. MaRS is built on ProMark, so the MaRS installation will require ProMark if it is not there.

- **Python**. Python is an interpreted 4GL language that is inherently open-source, fully documented and supported by a huge base of users. It is mainly used as a scripting tool in the MaRS environment. You will need the current Python version described in the MaRS setup. MaRS will not work without Python. MaRS users do not need to develop software with Python, but you may need to debug or enhance a Python script.

- **Git / GitHub**. Git is an open-source software development repository (**repo**) and versioning management tool suite. It can be used to manage source code, binaries, libraries, documents, images, etc. GitHub is an independent website that provides repository management facilities using Git. Intel uses GitHub as its repository facility. You will need to create and use an Intel GitHub account to access your work repos. You will also need to clone your target repos to your local PC where you will do your work. All tasks to clone, update, and share your work are done via Git commands.

- **BASH / GitBash**. BASH is an open-source Unix / Linux shell. Intel uses it for GitHub access. When you run the Git setup for MaRS, it will include GitBash, a command shell version that works on Windows and knows how to communicate with your GitHub repository. Another way to think about GitBash is that it is a Linux '**Terminal'**, offering a powerful character-based command language in a Windows command shell.

- **Markdown / Multi-Markdown**. ProMark uses an open-source language to author content for electronic publishing: **Markdown**. There is currently no ANSI standard for Markdown, but it is widely used with a common set of features. In its simplest form, Markdown is a pure text representation of HTML. Markdown does not use tags or embedded control characters. You create markdown files with ASCII text using a specific syntax and saved as text files with an **'.md'** extension. One of the basic rules of Markdown is that it can be readily converted to HTML without the use of tags or keywords. Markdown does have syntax supporting diagrams and images, animation and video, database calls, and external links, etc. The Markdown Guide is an easy-to-use reference for Markdown that is vendor and browser neutral: see https://www.markdownguide.org/.

  Multi-Markdown is a Markdown extension providing advanced publishing features like tables of contents, footnotes, scientific notation and chapter organization, etc. It is also unofficially standardized, but there are some vendor-specific implementations of Multi-Markdown. Multi-Markdown is also text-based and its file extension is **'.mmd'**. ProMark uses a Multi-Markdown extension. Consult this site for a browser and vendor neutral introduction to Multi-Markdown: https://fletcherpenney.net/multimarkdown/

- **Visual Studio Code (VSC).** Visual Studio Code is a freeware product from Microsoft. It is intended as an open-source tool that functions both as a powerful editing and scripting development environment for web publishing, and also as a 'Terminal' for communicating with Git / GitHub. It has built-in support for **Markdown** (MD) and Intel's **Multi-Markdown** (MMD) extension. It also has extensions for Python, Java, C++ and other languages. Visual Studio Code is not strictly required for ProMark / MaRS work. However, Visual Studio Code is the only editing platform that is directly supported by ProMark. If you want to use some other editor, that is fine as long as you use it in concert with Visual Studio Code.

  You use Visual Studio Code to author and publish your HAS content for MaRS development. Visual Studio Code processes the Markdown to generate HTML, using MaRS scripts that also generate other metadata files necessary for the process. Your work resides in your local cloned Git repository, which is later pushed to the GitHub repository master.

- **Excel / Visio**. When you install MaRS, it will check to make sure you have Excel and Visio installed. Excel should already be resident on your Intel PC as part of Microsoft Office. Visio may not be resident on your PC, but it is available in the Intel Software Market or the Intel Corporate Portal. The image processing that MaRS performs along the way uses Visio as its diagramming standard.

  Excel has a specific function in MaRS, which uses a proprietary **Excel plug-in** called ***ACES***. When you setup MaRS, it will require ACES installation. Intel uses Excel spreadsheets (.xlsx files) as a platform to manage interface templates. ProMark has a standard Reference Catalog of interfaces, which are maintained in Excel files. These topics will be covered in detail later in this document.

2. ProMark

As noted in the Prerequisites section, MaRS requires that ProMark be setup and working properly. When you start the MaRS setup, it will check and verify your

ProMark installation. If it is not resident on your target PC, it will install it. If there is some issue in your ProMark setup, you must get this resolved before proceeding. If necessary, contact Intel IT Support for assistance, using these resources:

http://it.intel.com
503-696-1234 (TAC USA hotline)

To install or update ProMark, use this Intel link:
https://docs.intel.com/documents/promark/GitHub/Github_pm_tools.html

You need to navigate to this site and follow the instructions to correctly install ProMark and its related tools. Download and run the install tool:
`ProMark_setup_github.exe.`

Use the checklist and described sequence to properly install ProMark. This will be followed by guidance to **Upgrade Python**. This document also provides information on the Python setup.

GitHub access is required, including an Intel GitHub account. You can use the unit's link to commence GitHub setup: GitHub onboarding process

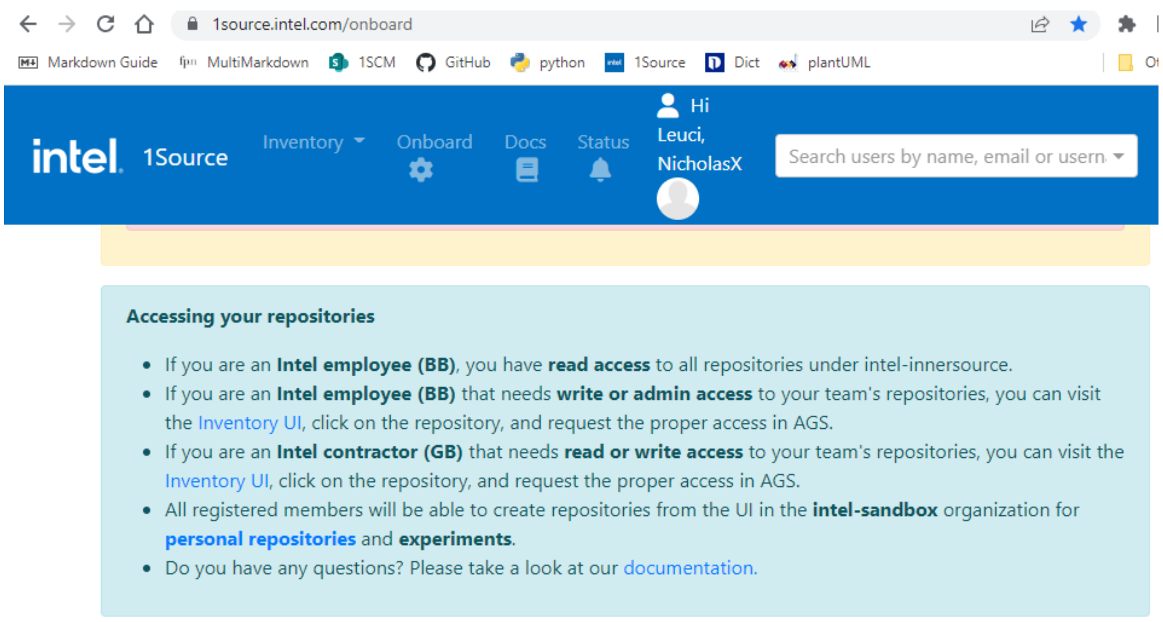
You will need to request AGS permissions for this, if you have not already done this. Your GitHub account status is shown in your Intel 1 Source page. These are non-trivial steps, and they must be fully and completely finished before you can use ProMark and MaRS. More information on GitHub is provided later in this document.

The most up-to-date information on ProMark itself can be found at this main site: http://goto/ProMark. This site has ALL the core components and references for ProMark. It is maintained and kept current.

There is another page that is the hub of ProMark tools and methodology uses. You can find this page at: https://docs.intel.com/documents/ProMark/index.html
Here you will find links to detailed information, instructional slide shows, video tutorials, and comprehensive documentation on the major aspects of ProMark development.

3. Build Hierarchy With GitHub / GitBash

The focus here is working on a cloned repo on a PC, under the assumption that the reader already has a current Intel GitHub login account with a GitHub SSO email account. These are necessary to be able to access and clone a development repository (repo) to your local PC. Contact your Intel manager if you don't have these accounts. You can always check your GitHub access via Intel [1Source](). Here is an example screen snap of the repo access summary shown at the bottom of your 1Source dashboard:



Accessing your repositories

- If you are an **Intel employee (BB)**, you have **read access** to all repositories under intel-innersource.
- If you are an **Intel employee (BB)** that needs **write or admin access** to your team's repositories, you can visit the Inventory UI, click on the repository, and request the proper access in AGS.
- If you are an **Intel contractor (GB)** that needs **read or write access** to your team's repositories, you can visit the Inventory UI, click on the repository, and request the proper access in AGS.
- All registered members will be able to create repositories from the UI in the **intel-sandbox** organization for **personal repositories** and **experiments**.
- Do you have any questions? Please take a look at our documentation.

When you have the authorizations and permissions to access your GitHub repo, the next step is to access and select it for cloning. There are several ways to clone a repo. This example here uses the HTTPS URL cloning method to select and copy the URL of a remote repo to your local PC. The cloned repo is a mirror image copy on your PC, allowing you to examine and work on the repo locally without directly affecting the source repo.

You will be required to sign-in to your GitHub account to access the remote repo. Here is a typical scenario to clone the Working HAS Demo (see the screen snap below:

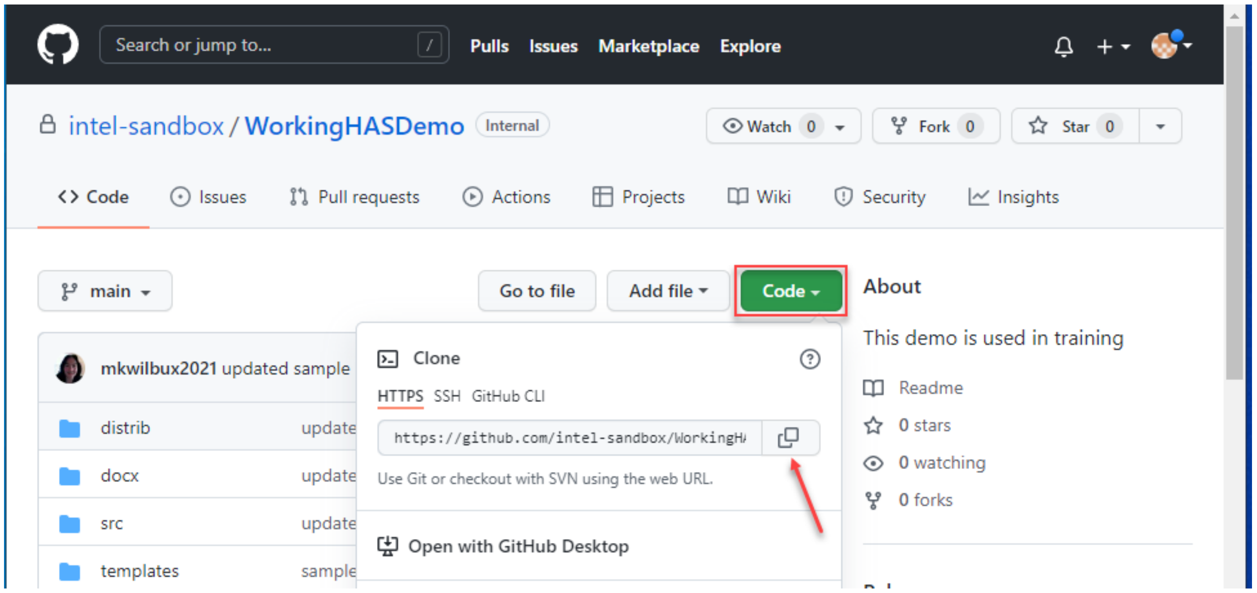
1. Copy this link to your browser and go there: https://github.com/intel-sandbox/WorkingHASDemo

2. Click **Code:**

3. Go to **Clone > HTTPS**

4. Click to copy the HTTPS URL to your PC's clipboard, as shown below.

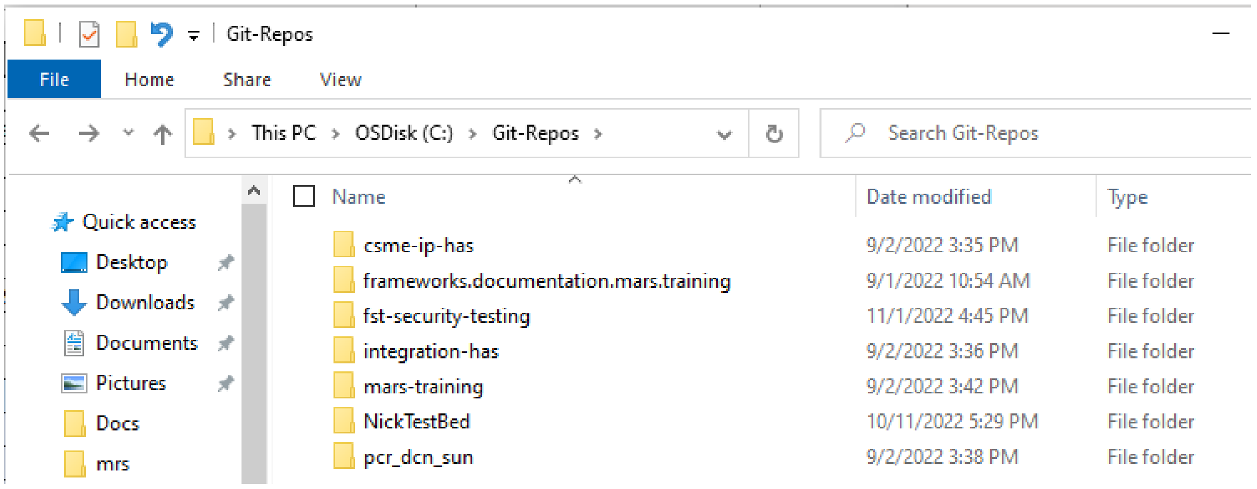
5. The URL link is '**Copied**' to your Windows clipboard.



At this point, your cloned repo is an image in your Windows clipboard. If you download a repo from GitHub using default settings, your local repo will be created relative to your Window's '**user**' settings. This is not an optimal working environment. A better approach is to create a dedicated working area for your repos.

You need to access your repo image n your Windows clipboard to clone it to your local PC. You can use Git Bash or Visual Studio Code's terminal to do this. The Git Bash terminal (or shell) is a command-line utility to perform computer operations. It does take input from the keyboard, but it also responds to mouse events (like button clicks) as well. It is like the MS PowerShell. However, the Git Bash implementation comes with 'hooks' (interfaces) pre-installed to communicate with Git, which makes it very useful for operating on Git / GitHub repos.

As mentioned, when you clone a repo, you need to create a local directory for your repo development. For example, on your PC create a directory called '**Git-Repos**' and use this as the root destination of your cloning operations. In the screen snap below, several repos have been cloned for local work:



Name	Date modified	Type
csme-ip-has	9/2/2022 3:35 PM	File folder
frameworks.documentation.mars.training	9/1/2022 10:54 AM	File folder
fst-security-testing	11/1/2022 4:45 PM	File folder
integration-has	9/2/2022 3:36 PM	File folder
mars-training	9/2/2022 3:42 PM	File folder
NickTestBed	10/11/2022 5:29 PM	File folder
pcr_dcn_sun	9/2/2022 3:38 PM	File folder

When you clone a repo, you also need to designate a named work area, which becomes a local directory for you to do your development work. You create this using a Git command in your Bash shell or your VSC Bash terminal. Position the shell at the root or top level location where you wish to place your local repo. Here, that location is '**`C:\Git-Repos`**'. Enter this paradigmatic Git command in your shell at the **`$`** prompt:

```
git clone <GitHub clone URL>
```

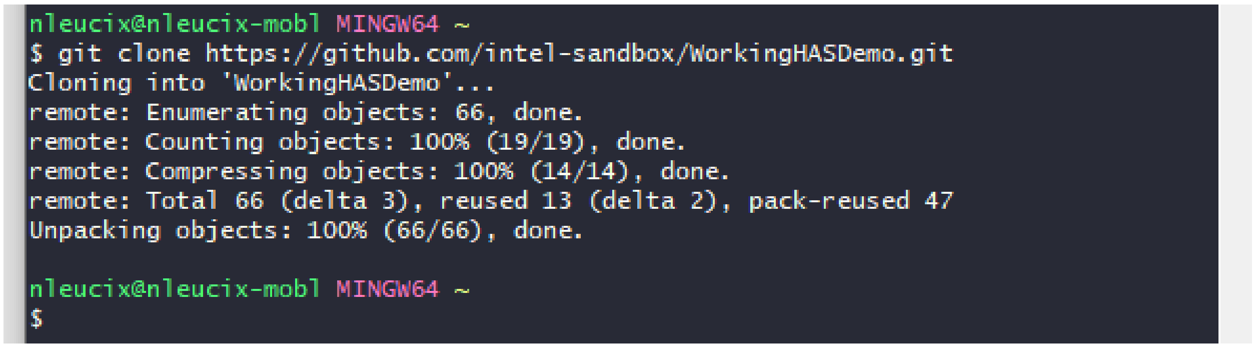
Immediately after hitting the 'Clone' button described above, your cloned URL is in the Windows clipboard. Then you use your Bash shell to navigate to the correct location and only type **`git clone`** and click Windows **paste** to fill in the URL string. The Bash shell will copy the cloned URL image to your selected navigation point on your PC, creating your local repo.

You can create your workspace (name or short name) by using that short name as the last argument of the **`git clone`** command, as shown below:

```
git clone https://intel-restricted/frameworks..(full URL path)
fst-security-testing
```

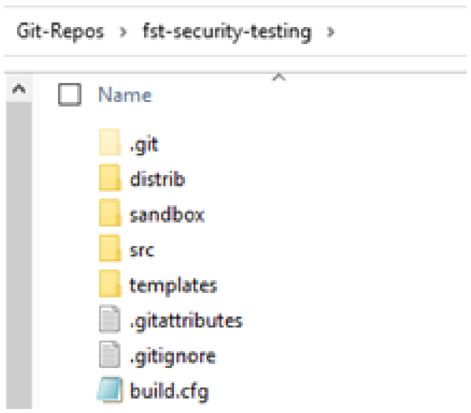
This will clone the repo into **`C:\Git-Repos\fst-security-testing.`**

Git will copy the repo to your selected work area, including all subdirectories and files. The screen snap below is an example of the Git output immediately after entering a **`git clone <GitHub clone URL>`** Bash command:



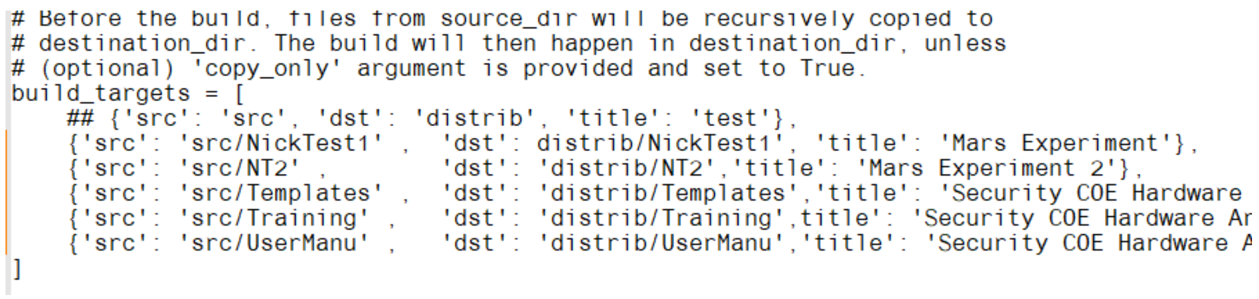
```
nleucix@nleucix-mobl MINGW64 ~
$ git clone https://github.com/intel-sandbox/WorkingHASDemo.git
Cloning into 'WorkingHASDemo'...
remote: Enumerating objects: 66, done.
remote: Counting objects: 100% (19/19), done.
remote: Compressing objects: 100% (14/14), done.
remote: Total 66 (delta 3), reused 13 (delta 2), pack-reused 47
Unpacking objects: 100% (66/66), done.

nleucix@nleucix-mobl MINGW64 ~
$
```

Each of these cloned repos are at the top level of a **working tree hierarchy**. When you perform various operations in them, Git knows how to synchronize the GitHub repo with your local working repo. Common tasks include `git pull` to download updated repo content to your tree, and `git push` to upload your local repo updated content to the GitHub master. These commands are executed from either a GitBash shell or your Visual Studio Code (VSC) terminal at the top level of your tree. In the examples shown in this Guide, the top-level tree hierarchy is a cloned repo called '**`fst-security-testing'`**, as shown below.



```
Git-Repos > fst-security-testing >
Name
  .git
  distrib
  sandbox
  src
  templates
  .gitattributes
  .gitignore
  build.cfg
```

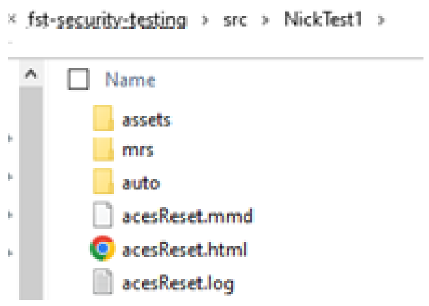
These directories were created during the cloning operation, along with your local work area which is a subdirectory named '**`NickTest1`**' in this Guide, Note the text file '**`build.cfg`**' in the root of the repo. This is your repo's **configuration file**. When you clone a repo and set up a working directory or add a new work area, you need to add an entry to this file so that the build scripts can find your work area. The screen snap below is an excerpt image showing two added work areas, here **`'src/NickTest1'`** and '**`src/NT2'`**. These two lines were added by editing this file. This will enable the script compile commands '**`bulld'`** and **`'iparch-build'`** to construct your application.



```
# Before the build, files from source_dir will be recursively copied to
# destination_dir. The build will then happen in destination_dir, unless
# (optional) 'copy_only' argument is provided and set to True.
build_targets = [
    ## {'src': 'src', 'dst': 'distrib', 'title': 'test'},
    {'src': 'src/NickTest1' ,  'dst': distrib/NickTest1', 'title': 'Mars Experiment'},
    {'src': 'src/NT2' ,        'dst': 'distrib/NT2','title': 'Mars Experiment 2'},
    {'src': 'src/Templates' ,  'dst': 'distrib/Templates','title': 'Security COE Hardware
    {'src': 'src/Training' ,   'dst': 'distrib/Training',title': 'Security COE Hardware Ar
    {'src': 'src/UserManu' ,   'dst': 'distrib/UserManu','title': 'Security COE Hardware A
]
```

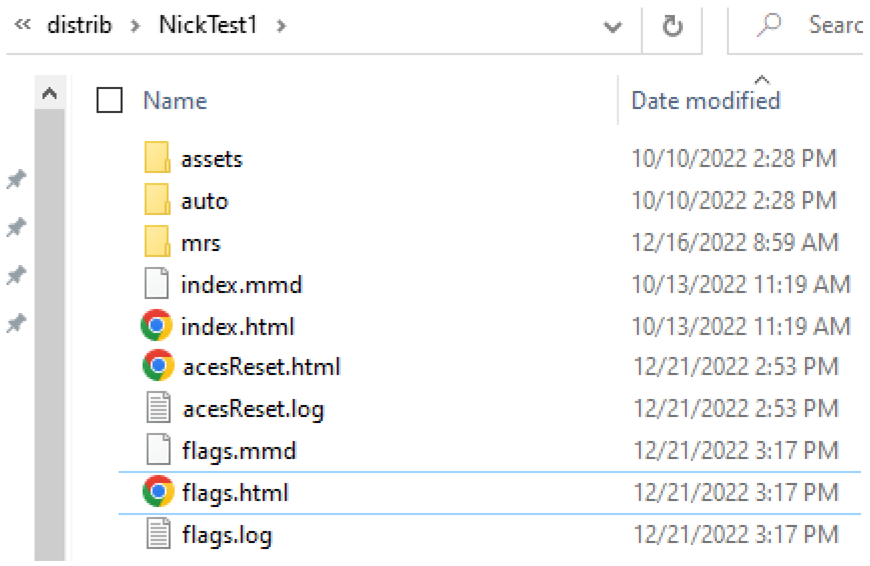
The two top-level key working directories are **`src`** and **`distrib`**. In general, the **`src`** subtree holds the inputs or sources for your development work, and the **`distrib`** subtree holds the outputs or results of the build operations from your local development.

Navigate to your **`src/`*`localworkdir`*** subtree to create your markdown files in VSC for your builds. Save your MaRS work as **'.mmd'** files here. See the screen snap below.



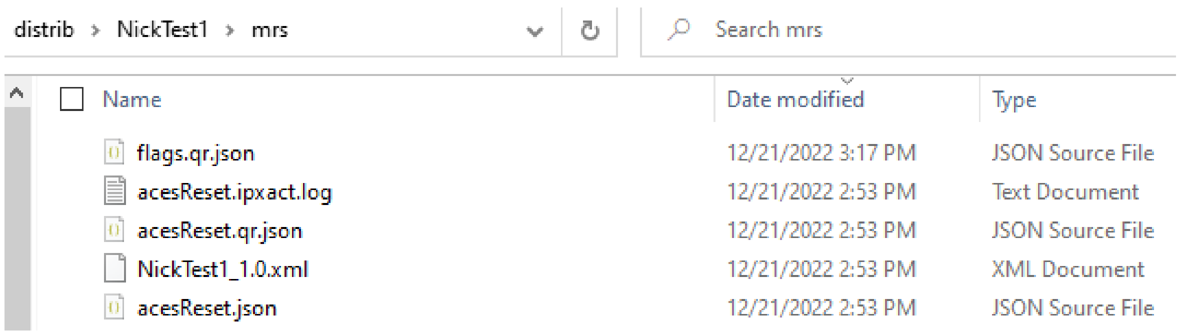
You will need to access and customize Excel templates as '.xlsx' files to capture your signal data, and you can save these templates in your local working directory or in the '**`assets`**' subdirectory. Acquiring, modifying, and building from these templates are covered in more detail later in this Guide.

The **`distrib`** subtree also has a directory with the name of your local work area, which is **`NickTest1`** in the screen snap shown below:



Name	Date modified
assets	10/10/2022 2:28 PM
auto	10/10/2022 2:28 PM
mrs	12/16/2022 8:59 AM
index.mmd	10/13/2022 11:19 AM
index.html	10/13/2022 11:19 AM
acesReset.html	12/21/2022 2:53 PM
acesReset.log	12/21/2022 2:53 PM
flags.mmd	12/21/2022 3:17 PM
flags.html	12/21/2022 3:17 PM
flags.log	12/21/2022 3:17 PM

This working area holds the outputs or results of your build operations. In the example above, the generated output was an HTML file and related log and ancillary files. The **`'mrs'`** subdirectory shown below is specific to MaRS / ACES functionality for capturing the output XML and related JSON outputs.



Name	Date modified	Type
flags.qr.json	12/21/2022 3:17 PM	JSON Source File
acesReset.ipxact.log	12/21/2022 2:53 PM	Text Document
acesReset.qr.json	12/21/2022 2:53 PM	JSON Source File
NickTest1_1.0.xml	12/21/2022 2:53 PM	XML Document
acesReset.json	12/21/2022 2:53 PM	JSON Source File

There are two basic ProMark commands to build multi-markdown content. The first method is to do an **immediate build**. You do this by navigating to your source editing area in the repo **`'src/yourlocaldir/'`** in your VSC terminal and entering this command: '**`mmd2doc yourworkfilename.mmd --chrome'`**. This command invokes a Python script that converts your markdown '.mmd' file into an HTML file placed in the **`distrib/yourlocaldir`** output location. The **`'--chrome'`** argument to the command is optional, allowing the HTML to also be rendered in your Chrome browser immediately. If there is a defect in your build, appropriate error messages will be generated in your VSC terminal, allowing you to address each defect and then rebuild accordingly.

The second build method is the **repo-level build**. In your VSC terminal, navigate to the top level of your build hierarchy; in our example this is **`fst-security-`**

`testing`. Enter this command: **`build`**. This rebuilds all development sources in the repo, if they are not current, and generates an '**`index.html`**' file in your **`distrib/localworkdir`** location. When you click on this HTML file, it opens it in your browser, and it will show clickable display links to your work output(s).

Note that later in this Guide, the IPArch container setup is described (in Section 5). When the IPArch container is installed, new wrapper commands are used to do your builds. They are:

- **`'iparch-mmd2doc yourworkfilename.mmd --chrome'`**
  (local or immediate builds)
- **`'iparch-build'`**
  (repo-level or global builds)

The implicit assumption here is that users will invoke Visual Studio Code (VSC) from the installed Windows program icon and run these commands in VSC's Bash terminal. An alternative is to invoke VSC directly from your Bash shell and use keyboard macros to build VSC '.mmd' applications and display them in a browser. How to do this is described later in section 7, Application Example.

The work you do in your local repo is intended to be folded back into the GitHub master repo it was cloned from. This can include your new source files, your working Excel templates, subdirectories with diagrams or other documents, etc. Before you synchronize your repo with the master repo, save all your work and remove any scratch or temporary files or directories you do not want to retain. Always do a **`git pull`** BEFORE pushing your work back.

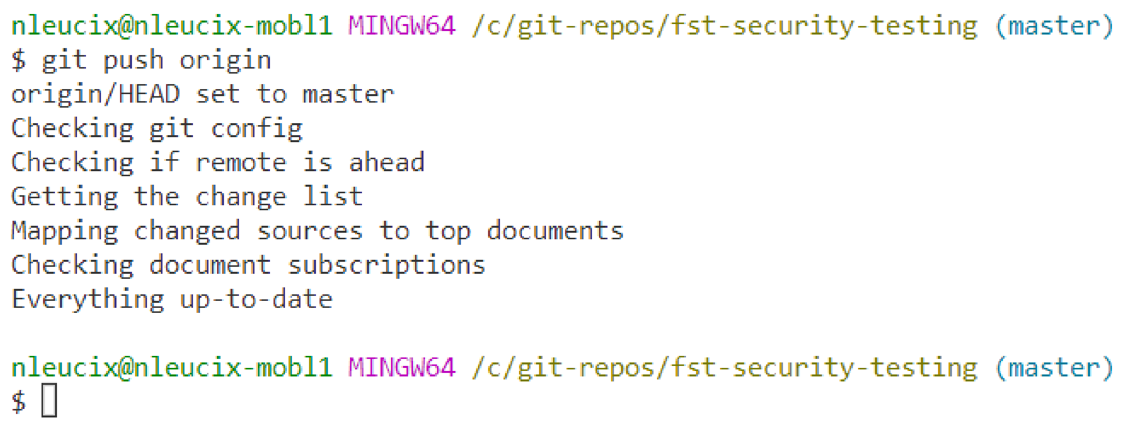
There are several Git commands to push your work back to the master repo. You always need to **save** your source files, **`add`** and then **`commit`** them to your local repo **FIRST** before doing a push. Here are some useful and common Git commands to run in your VSC terminal:

```
git status                       (Get current Git status)
git pull                         (Update your ProMark environment)
git add yoursourcefile           (Each changed file must be fully named)
git commit -m "your comments"    (Each changed file must be fully named)
git push                         (Synchronize back to the master repo)
```

Also, this command is useful to execute:

```
git push origin                  (Do this at your local repo's top-level)
git push origin localworkdir     (Do this at your local work area)
```

Here is an example of a Git **`origin push`**, done in the VSC terminal after all **`git add`** and **`git commit`** commands were successfully run:



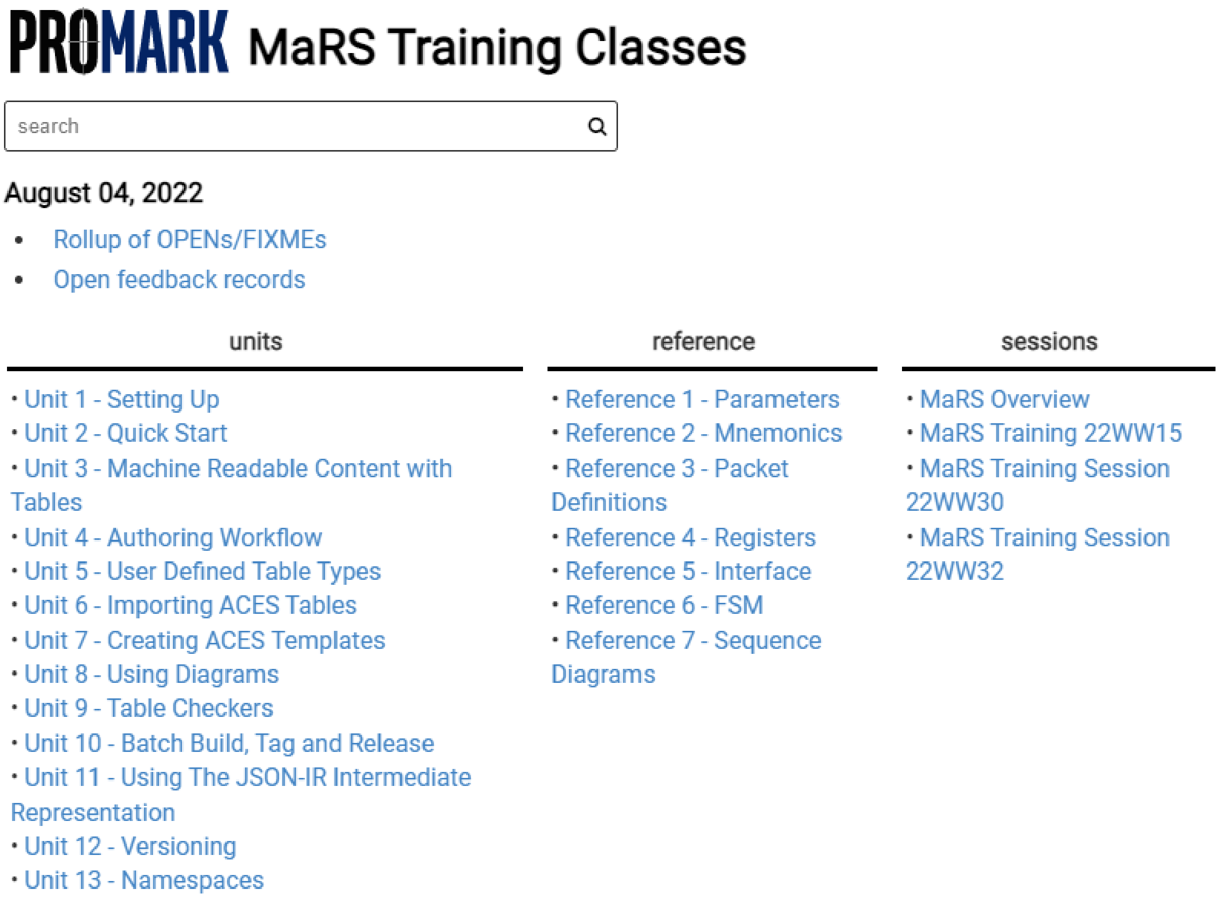
```
nleucix@nleucix-mobl1 MINGW64 /c/git-repos/fst-security-testing (master)
$ git push origin
origin/HEAD set to master
Checking git config
Checking if remote is ahead
Getting the change list
Mapping changed sources to top documents
Checking document subscriptions
Everything up-to-date

nleucix@nleucix-mobl1 MINGW64 /c/git-repos/fst-security-testing (master)
$ 
```

At this point all local repo work has been synchronized and updated back to the GitHub repo from which the local repo was cloned. You work is then available to all with permissions to access your GitHub master repo.

4. MaRS

There is an internal site dedicated to ProMark MaRS training, and this is the best place to start using MaRS:
https://docs.intel.com/documents/il-specs/MaRS/training/classes/index.html

As shown below, this site is structured in numbered training **units**, with additional **reference** pages. The first-time user should visit the units in order and perform the exercises and attempt to replicate the examples. This content is challenging and robust, but your time will be well spent. Please note that MaRS and its training site are a developing work-in-progress.



PROMARK MaRS Training Classes

search

August 04, 2022

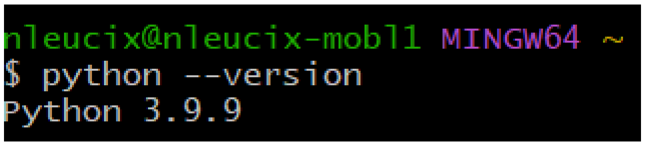
- Rollup of OPENs/FIXMEs
- Open feedback records

units	reference	sessions
• Unit 1 - Setting Up	• Reference 1 - Parameters	• MaRS Overview
• Unit 2 - Quick Start	• Reference 2 - Mnemonics	• MaRS Training 22WW15
• Unit 3 - Machine Readable Content with Tables	• Reference 3 - Packet Definitions	• MaRS Training Session 22WW30
• Unit 4 - Authoring Workflow	• Reference 4 - Registers	• MaRS Training Session 22WW32
• Unit 5 - User Defined Table Types	• Reference 5 - Interface	
• Unit 6 - Importing ACES Tables	• Reference 6 - FSM	
• Unit 7 - Creating ACES Templates	• Reference 7 - Sequence Diagrams	
• Unit 8 - Using Diagrams		
• Unit 9 - Table Checkers		
• Unit 10 - Batch Build, Tag and Release		
• Unit 11 - Using The JSON-IR Intermediate Representation		
• Unit 12 - Versioning		
• Unit 13 - Namespaces		

Click on Unit 1 to start the ProMark setup, using the unit's provided link: https://docs.intel.com/documents/il-specs/mars/training/classes/units/1_Setting_Up.html#:~:text=https%3A//docs.intel.com/documents/promark/GitHub/Github_pm_tools.html.

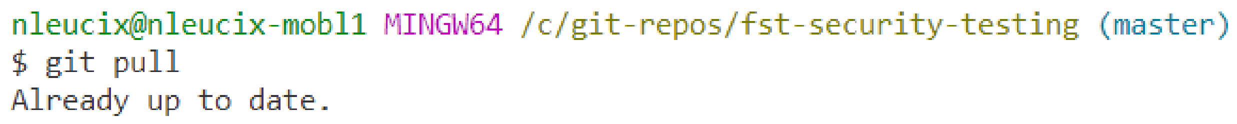
Python

The ProMark setup will check the Python installation. Your version should be 3.9 or better. You can easily check your Python version in a Windows command shell or the Bash shell by running the command '**`python --version`**', as shown below:



```
nleucix@nleucix-mobl1 MINGW64 ~
$ python --version
Python 3.9.9
```

You need to regularly check to make sure your ProMark / MaRS environment is up to date. The best way to do this is to navigate to the top of your repo's development tree in your BASH shell or VSC terminal and at the **$** prompt enter this git command: **`git pull`**. It will update your development environment or report that you are currently up to date, as shown below.



```
nleucix@nleucix-mobl1 MINGW64 /c/git-repos/fst-security-testing (master)
$ git pull
Already up to date.
```

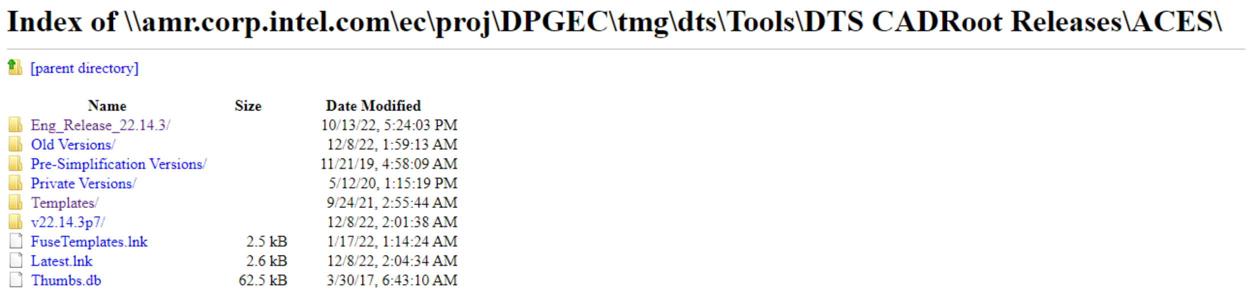
The Git repo hierarchy for ProMark / MaRS is described in the previous chapter (Section 3).

ACES

Once Python is installed, the next step is the ACES setup. ACES is an internal **plug-in for Excel** developed specifically for ProMark and MaRS use. It is important to note that these tools, and ACES specifically, are moving targets. They are versioned and maintained by Intel. MaRS users should periodically check the released version and always have the current version of ACES installed.

Use this link to navigate to the Intel ACES release site:
`\\amr.corp.intel.com\ec\proj\DPGEC\tmg\dts\Tools\DTS CADRoot Releases\ACES\`

Here is a recent listing:



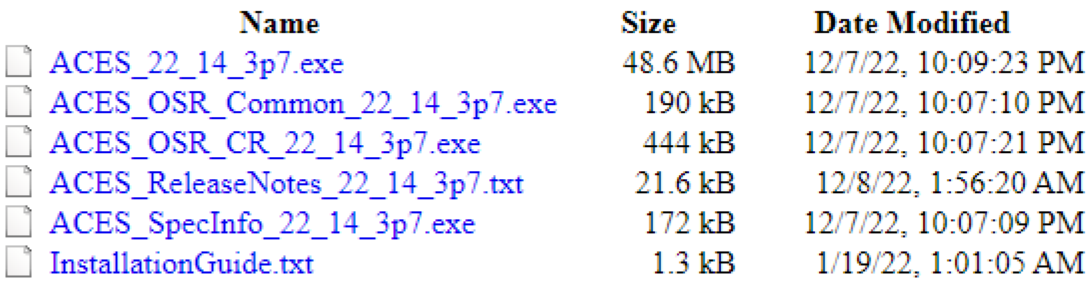
Index of \\amr.corp.intel.com\ec\proj\DPGEC\tmg\dts\Tools\DTS CADRoot Releases\ACES\

[parent directory]

Name	Size	Date Modified
Eng_Release_22.14.3/		10/13/22, 5:24:03 PM
Old Versions/		12/8/22, 1:59:13 AM
Pre-Simplification Versions/		11/21/19, 4:58:09 AM
Private Versions/		5/12/20, 1:15:19 PM
Templates/		9/24/21, 2:55:44 AM
v22.14.3p7/		12/8/22, 2:01:38 AM
FuseTemplates.lnk	2.5 kB	1/17/22, 1:14:24 AM
Latest.lnk	2.6 kB	12/8/22, 2:04:34 AM
Thumbs.db	62.5 kB	3/30/17, 6:43:10 AM

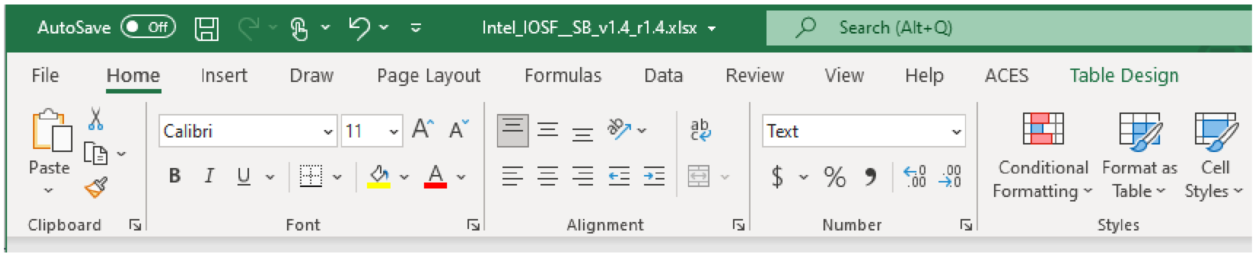
If you need to upgrade ACES on your PC, go to the Windows **Control Panel** and select the **Programs** feature to completely uninstall ACES before re-running the ACES installation.

As of this writing, the current version of ACES is **v22.14.3p7**. Click on the current link to see the installation contents, as shown here:

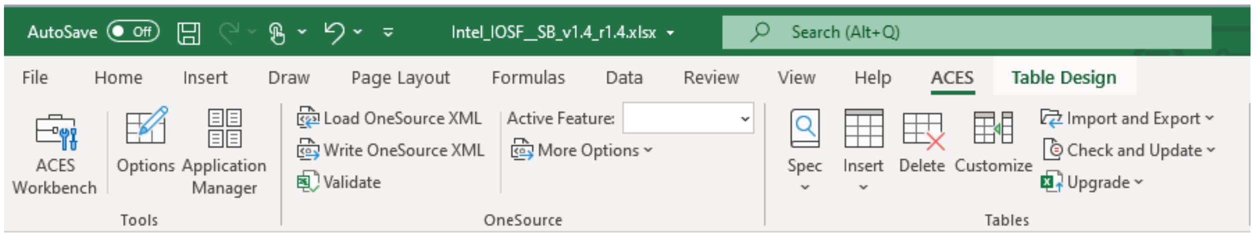


Name	Size	Date Modified
ACES_22_14_3p7.exe	48.6 MB	12/7/22, 10:09:23 PM
ACES_OSR_Common_22_14_3p7.exe	190 kB	12/7/22, 10:07:10 PM
ACES_OSR_CR_22_14_3p7.exe	444 kB	12/7/22, 10:07:21 PM
ACES_ReleaseNotes_22_14_3p7.txt	21.6 kB	12/8/22, 1:56:20 AM
ACES_SpecInfo_22_14_3p7.exe	172 kB	12/7/22, 10:07:09 PM
InstallationGuide.txt	1.3 kB	1/19/22, 1:01:05 AM

Close all instances of Excel before running the ACES installations. Click on each ACES component to open, download and install it. You need ALL of these components. Start with the Installation guide **`InstallationGuide.txt`**. Read and **follow the instructions in order.** When you run each installation component, it will operate in a Windows command shell, which will close when the procedure finishes. When ACES is fully installed, you will see the top-level Menu for the ACES plug-in in the top of the ribbon whenever you launch Excel, as shown below:



If you click on the ACES Menu, Excel will present the ACES plug-in functionality, which should resemble the following ribbon:



The uses of some of these ACES features are covered in the MaRS training Units 6 and 7 and throughout the training course.

Visual Studio Code (VSC)

Visual Studio Code is a Microsoft markdown editor and Bash terminal for development in a Git environment. It has several features that make it essential:

- It has built-in support for Markdown and its extensions as an easy-to-use GUI program
- It can easily show / convert Markdown to HTML
- It has built-in support for Git / GitHub and it has a built-in Bash terminal that understands Git commands
- It has support and language extensions / packages for a wide variety of programming languages, including Python, Java, C++, and many others.
- It is free and open-source, with a large internal support group in Microsoft and a huge developer community around the world.
- Many other VSC extensions are available for various languages and other related tools and functionalities.

ProMark does not require the installation of Visual Studio Code (VSC). However, you do need a tool that has all its features. However, the IPArch container described later in this document does assume that you are using VSC. As a practical matter, this document assumes that VSC is required. Note that Microsoft also has Markdown versions for the Mac and Ubuntu Linux. You can also additionally use any other editor or Markdown tool(s) of your choice, such as Notepad++ or VIM etc.

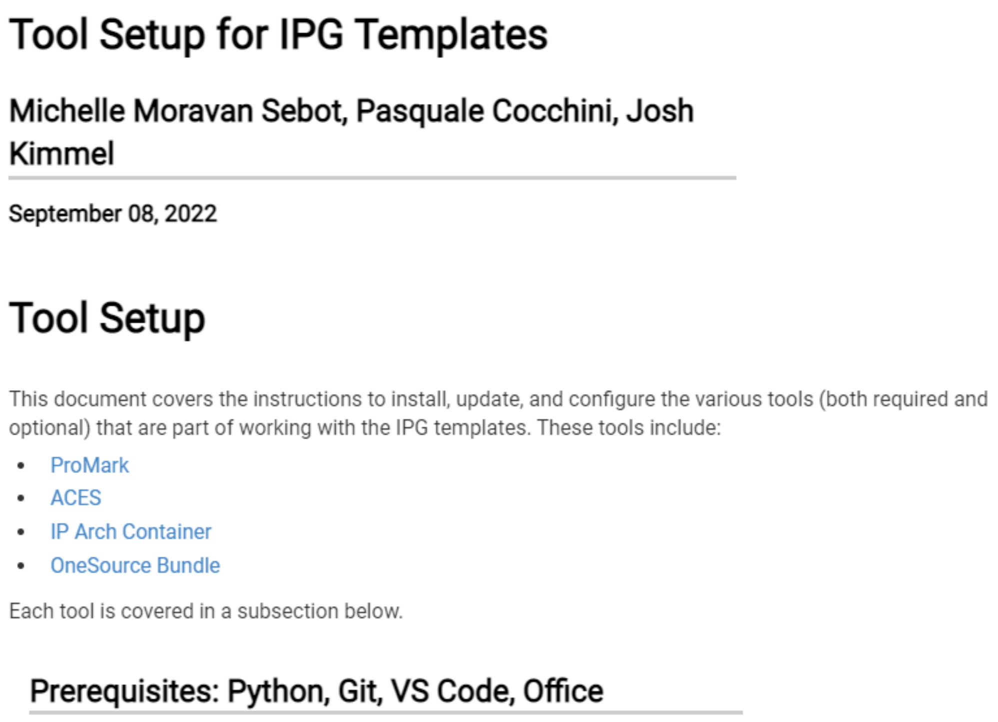
Here is the public link Microsoft provides for downloading VSC:
https://code.visualstudio.com/

The MaRS Unit 1 does have links and suggestions for 'Useful Extensions'. These are not required for ProMark or MaRS but you might find them to be useful.

5. IPArch Container

Setting up ProMark, GitHub, Visual Studio Code and ACES are essential building blocks to build a useful platform. One final implementation step is necessary for MaRS development. That step is setting up the IPArch container.

Detailed instructions for setting up the IPArch container and related tools are presented in this page, which you should bookmark:
https://docs.intel.com/documents/iparch/iptech/Documentation/ToolSetup.html



Tool Setup for IPG Templates

Michelle Moravan Sebot, Pasquale Cocchini, Josh Kimmel

September 08, 2022

Tool Setup

This document covers the instructions to install, update, and configure the various tools (both required and optional) that are part of working with the IPG templates. These tools include:

- ProMark
- ACES
- IP Arch Container
- OneSource Bundle

Each tool is covered in a subsection below.

Prerequisites: Python, Git, VS Code, Office

The guidance provided should be followed carefully. Make sure that ALL the described prerequisites are installed, current and working properly. Most of this discussion has already been covered here and elsewhere, but it is useful to revisit it.

ProMark

One key provision is the guidance on maintaining the **CURRENT** ProMark version. As noted, ProMark is a moving target. You need to regularly check to be sure your version is current. The discussion on ProMark contains many useful links,

ACES

The coverage of ACES here is more detailed than in the MaRS training course. If you have any difficulty with ACES, revisit this section. The following excerpt is key: ”The ACES tool will also prompt users to update to a newer version, when available, through a pop-up message. In that case, click yes and follow instructions to update your ACES installation. This typically involves first uninstalling the existing version and then installing the newer version per the instructions above.”

All the prerequisites need to be correctly implemented before installing the IPArch Container.

This IPArch container sets up the development environment with the use of wrapper commands. The instructions are complex. We revisit them in detail here in the interests of clarity.

You MUST use the Git Bash shell for these instructions. Do not use the Windows command prompt or PowerShell for this set up.

1, Launch the Git Bash shell. Only the first time, or whenever Git Bash changes, enable the local installation of the templates in **<u>your</u>** GIT bash shell. The following command is one line of script command. Cut and paste this entire command string at the **$** prompt and press ***Enter***:

//IPACSHARE-DM.cps.intel.com/ipac/mrs/templates/ip_template/releases/latest/setup.sh

2. For the changes to the shell to take place, either close and reload a new Git Bash shell, or source the IPARCH specific settings:

```
$ source ~/.bashrc-iparch
```

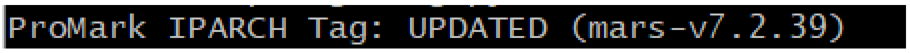
Again, note that you only need to do this the first time or whenever your Git Bash installation changes.

3. To complete the container installation, simply call the now available **`iparch-update`** Bash command to load the most up-to-date release:

```
$ iparch-update
IP_Template Update: v2.2.0
Archive: //IPACSHARE-DM.cps.intel.com/ipac/.../releases/latest/IP_Template-2.2.0.zip
<...>
IP_Template Update: SUCCESS
```

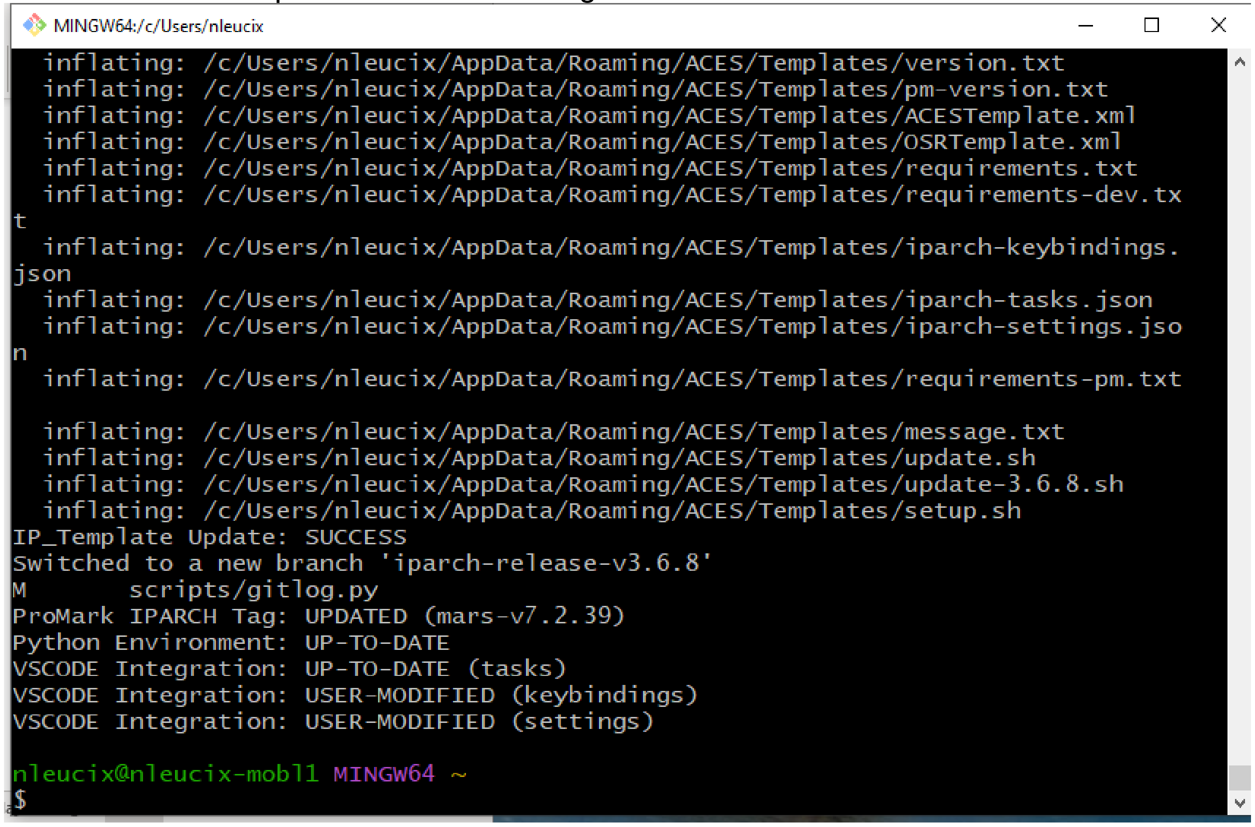
4. You will need to call **`iparch-update`** whenever there is a need for installing updates, depending on the policy set in place. Always keep your template package up to date by frequently calling **`iparch-update`** in your Bash shell. The IPArch team may make channel announcements with further instructions to update. When there is a new update, it will take a minute or so to install it in your IPArch container environment. Your Bash shell will show status messages while processing until it finishes with '**`Success`**'.

When the update output completes, it will have included the **current MaRS version**, as shown here:



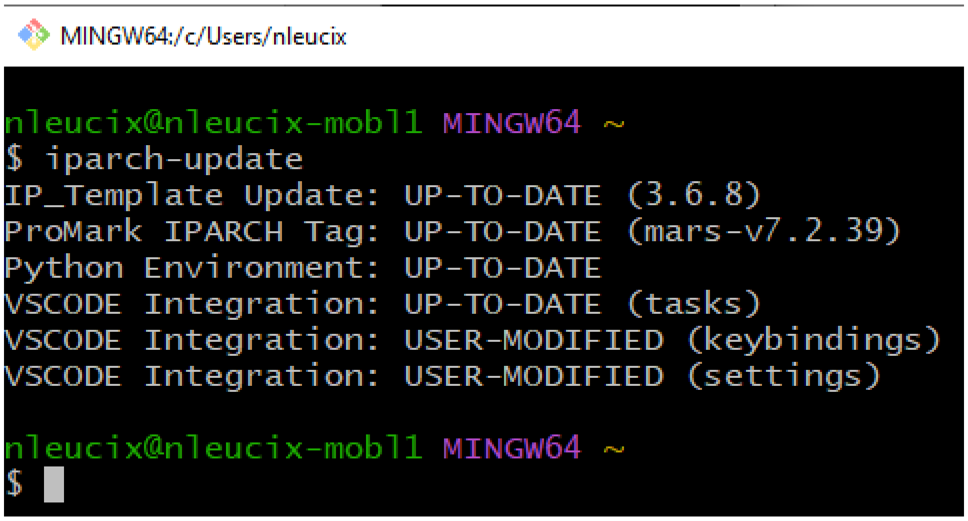
```
ProMark IPARCH Tag: UPDATED (mars-v7.2.39)
```

The final status output will look something like this:



```
  inflating: /c/Users/nleucix/AppData/Roaming/ACES/Templates/version.txt
  inflating: /c/Users/nleucix/AppData/Roaming/ACES/Templates/pm-version.txt
  inflating: /c/Users/nleucix/AppData/Roaming/ACES/Templates/ACESTemplate.xml
  inflating: /c/Users/nleucix/AppData/Roaming/ACES/Templates/OSRTemplate.xml
  inflating: /c/Users/nleucix/AppData/Roaming/ACES/Templates/requirements.txt
  inflating: /c/Users/nleucix/AppData/Roaming/ACES/Templates/requirements-dev.tx
t
  inflating: /c/Users/nleucix/AppData/Roaming/ACES/Templates/iparch-keybindings.
json
  inflating: /c/Users/nleucix/AppData/Roaming/ACES/Templates/iparch-tasks.json
  inflating: /c/Users/nleucix/AppData/Roaming/ACES/Templates/iparch-settings.jso
n
  inflating: /c/Users/nleucix/AppData/Roaming/ACES/Templates/requirements-pm.txt

  inflating: /c/Users/nleucix/AppData/Roaming/ACES/Templates/message.txt
  inflating: /c/Users/nleucix/AppData/Roaming/ACES/Templates/update.sh
  inflating: /c/Users/nleucix/AppData/Roaming/ACES/Templates/update-3.6.8.sh
  inflating: /c/Users/nleucix/AppData/Roaming/ACES/Templates/setup.sh
IP_Template Update: SUCCESS
Switched to a new branch 'iparch-release-v3.6.8'
M       scripts/gitlog.py
ProMark IPARCH Tag: UPDATED (mars-v7.2.39)
Python Environment: UP-TO-DATE
VSCODE Integration: UP-TO-DATE (tasks)
VSCODE Integration: USER-MODIFIED (keybindings)
VSCODE Integration: USER-MODIFIED (settings)

nleucix@nleucix-mobl1 MINGW64 ~
$
```

5. Run **`iparch-update`** daily. It is possible to have multiple updates in the same day. When your container is current and no update is needed, the Bash shell will show this kind of output:

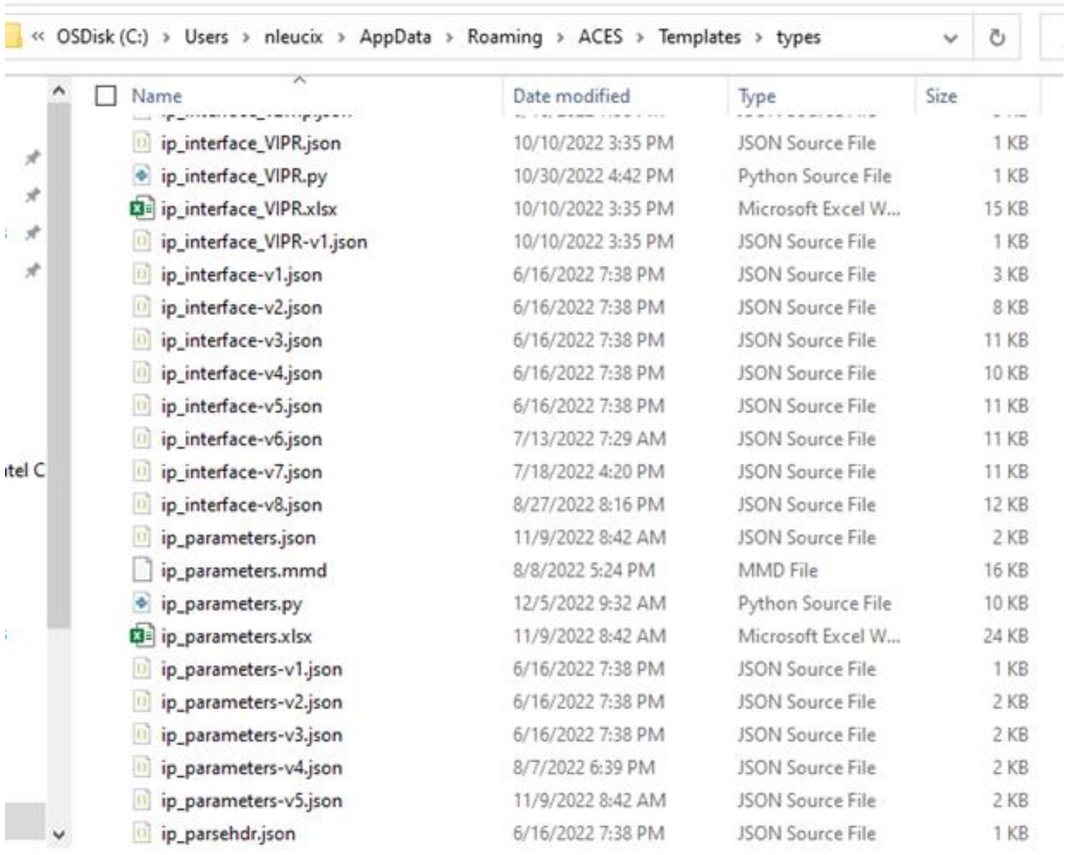


```
nleucix@nleucix-mobl1 MINGW64 ~
$ iparch-update
IP_Template Update: UP-TO-DATE (3.6.8)
ProMark IPARCH Tag: UP-TO-DATE (mars-v7.2.39)
Python Environment: UP-TO-DATE
VSCODE Integration: UP-TO-DATE (tasks)
VSCODE Integration: USER-MODIFIED (keybindings)
VSCODE Integration: USER-MODIFIED (settings)

nleucix@nleucix-mobl1 MINGW64 ~
$
```

When you install and update the IPArch container, it reconfigures the MaRS development environment. Among other tasks, it performs the following:

- Modifies the tools paths
- Sets up new Bash Shell commands, each with the '**`iparch-`**' prefix
- Provides the latest ACES template types (JSON files)

The IPArch container does not remove the original directory of JSON types set up by ACES. That file directory is still resident on your PC. It does set up a **new directory of JSON types** with the latest versions of all templates. Refer to the screen snap example below, which shows the fully expanded path to this new directory of JSON files. Note the iterations of **`ip_interface`** and **`ip_parameters`** files. You do not need to edit these files. The IPArch container alters your ACES path to this location to find the needed JSON types.



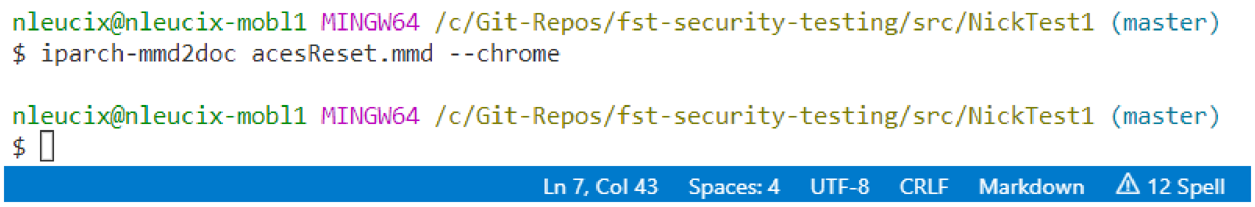
« OSDisk (C:) › Users › nleucix › AppData › Roaming › ACES › Templates › types

Name	Date modified	Type	Size
ip_interface_VIPR.json	10/10/2022 3:35 PM	JSON Source File	1 KB
ip_interface_VIPR.py	10/30/2022 4:42 PM	Python Source File	1 KB
ip_interface_VIPR.xlsx	10/10/2022 3:35 PM	Microsoft Excel W...	15 KB
ip_interface_VIPR-v1.json	10/10/2022 3:35 PM	JSON Source File	1 KB
ip_interface-v1.json	6/16/2022 7:38 PM	JSON Source File	3 KB
ip_interface-v2.json	6/16/2022 7:38 PM	JSON Source File	8 KB
ip_interface-v3.json	6/16/2022 7:38 PM	JSON Source File	11 KB
ip_interface-v4.json	6/16/2022 7:38 PM	JSON Source File	10 KB
ip_interface-v5.json	6/16/2022 7:38 PM	JSON Source File	11 KB
ip_interface-v6.json	7/13/2022 7:29 AM	JSON Source File	11 KB
ip_interface-v7.json	7/18/2022 4:20 PM	JSON Source File	11 KB
ip_interface-v8.json	8/27/2022 8:16 PM	JSON Source File	12 KB
ip_parameters.json	11/9/2022 8:42 AM	JSON Source File	2 KB
ip_parameters.mmd	8/8/2022 5:24 PM	MMD File	16 KB
ip_parameters.py	12/5/2022 9:32 AM	Python Source File	10 KB
ip_parameters.xlsx	11/9/2022 8:42 AM	Microsoft Excel W...	24 KB
ip_parameters-v1.json	6/16/2022 7:38 PM	JSON Source File	1 KB
ip_parameters-v2.json	6/16/2022 7:38 PM	JSON Source File	2 KB
ip_parameters-v3.json	6/16/2022 7:38 PM	JSON Source File	2 KB
ip_parameters-v4.json	8/7/2022 6:39 PM	JSON Source File	2 KB
ip_parameters-v5.json	11/9/2022 8:42 AM	JSON Source File	2 KB
ip_parsehdr.json	6/16/2022 7:38 PM	JSON Source File	1 KB

Here are the two most common IPArch container shell commands to run in the VSC terminal (which is also a Bash shell).

1. At your VSC MMD work location use this command to do a local build:
   `Iparch-mmd2doc` ***`yourMMDfile --anyflagsorotherargs`***

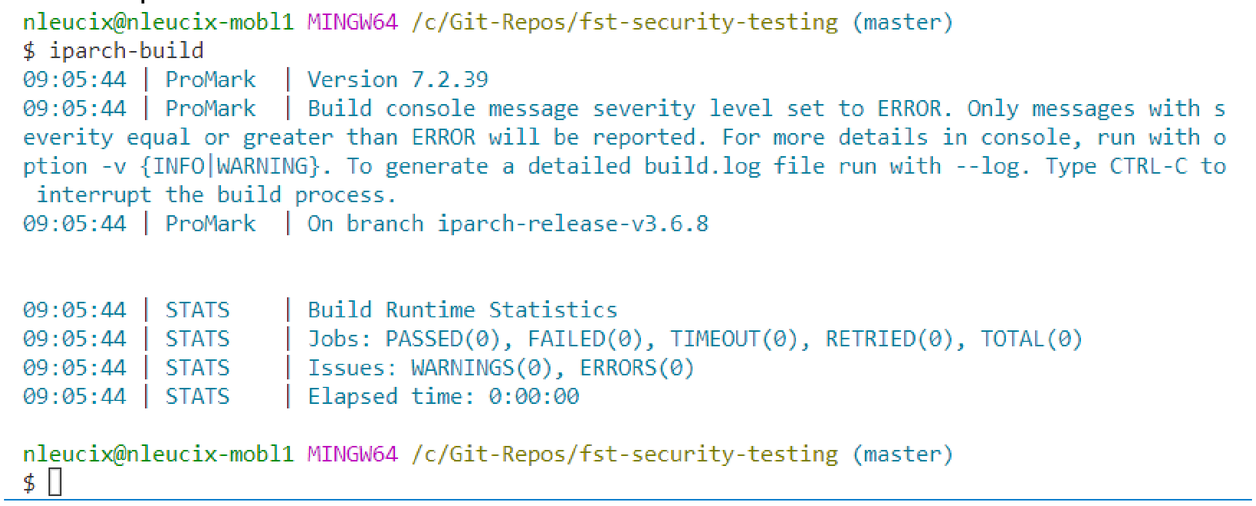
For example:



```
nleucix@nleucix-mobl1 MINGW64 /c/Git-Repos/fst-security-testing/src/NickTest1 (master)
$ iparch-mmd2doc acesReset.mmd --chrome

nleucix@nleucix-mobl1 MINGW64 /c/Git-Repos/fst-security-testing/src/NickTest1 (master)
$ 
```

2. At the root of your VSC development environment use this command to do a full build:
   `iparch-build`

For example:



```
nleucix@nleucix-mobl1 MINGW64 /c/Git-Repos/fst-security-testing (master)
$ iparch-build
09:05:44 | ProMark  | Version 7.2.39
09:05:44 | ProMark  | Build console message severity level set to ERROR. Only messages with s
everity equal or greater than ERROR will be reported. For more details in console, run with o
ption -v {INFO|WARNING}. To generate a detailed build.log file run with --log. Type CTRL-C to
 interrupt the build process.
09:05:44 | ProMark  | On branch iparch-release-v3.6.8


09:05:44 | STATS    | Build Runtime Statistics
09:05:44 | STATS    | Jobs: PASSED(0), FAILED(0), TIMEOUT(0), RETRIED(0), TOTAL(0)
09:05:44 | STATS    | Issues: WARNINGS(0), ERRORS(0)
09:05:44 | STATS    | Elapsed time: 0:00:00

nleucix@nleucix-mobl1 MINGW64 /c/Git-Repos/fst-security-testing (master)
$ 
```

Note both the ProMark Version and the IPArch Version in the build's output shown above.

<u>For further information</u>

This document is an installation and setup guide, and it is not intended as a full reference. The IPArch container and advanced MaRS / ACES development is a huge topic. If you need further information, the link below is a good place to take the next step:

https://docs.intel.com/documents/iparch/iptech/Documentation/IPArchContainer.html

Also, here is link for the ProMark reference page on IPArch and iptech templates:

https://docs.intel.com/documents/iparch/iptech/Templates/index.html

Please see below table for the list of common properties for the IPArch Container:

List of IPArch Repos	IPArch/*
Request View Access	[Preferred] AMR\IPArch - Document read BB AMR\IPArch - Document read GB [Old] AMR\ProductEngineering_PEG_FuncDesign_1274_10nm AMR\ProductEngineering_PEG_FuncDesign_Current_IRS AMR\ProductEngineering_PEG_FuncDesign_Future_ITS AMR\Promark - IP Architecture Group HAS GB
Request Publish Access	[Preferred] AMR\IPArch - Document write [Old] AMR\CICG IP Architecture AMR\IPArch_doc_write
Main Owner/Request Access	Ryan Holmqvist
GitHub Repo Access	Read: AGS "iparch all read" Write: AGS "github iparch maintain"
Size	<1.1GB
Release	Manual
DDP Location	https://docs.intel.com/documents/iparch/

OneSource Bundle

The OneSource Bundle contains utilities that assist with the specification of registers. It is integrated with the other tools described here. Register specification is done in a single framework while still being included in produced architecture documents. For general information, refer to the OneSource Wiki.

Please refer to the OneSource documentation set up on Windows.

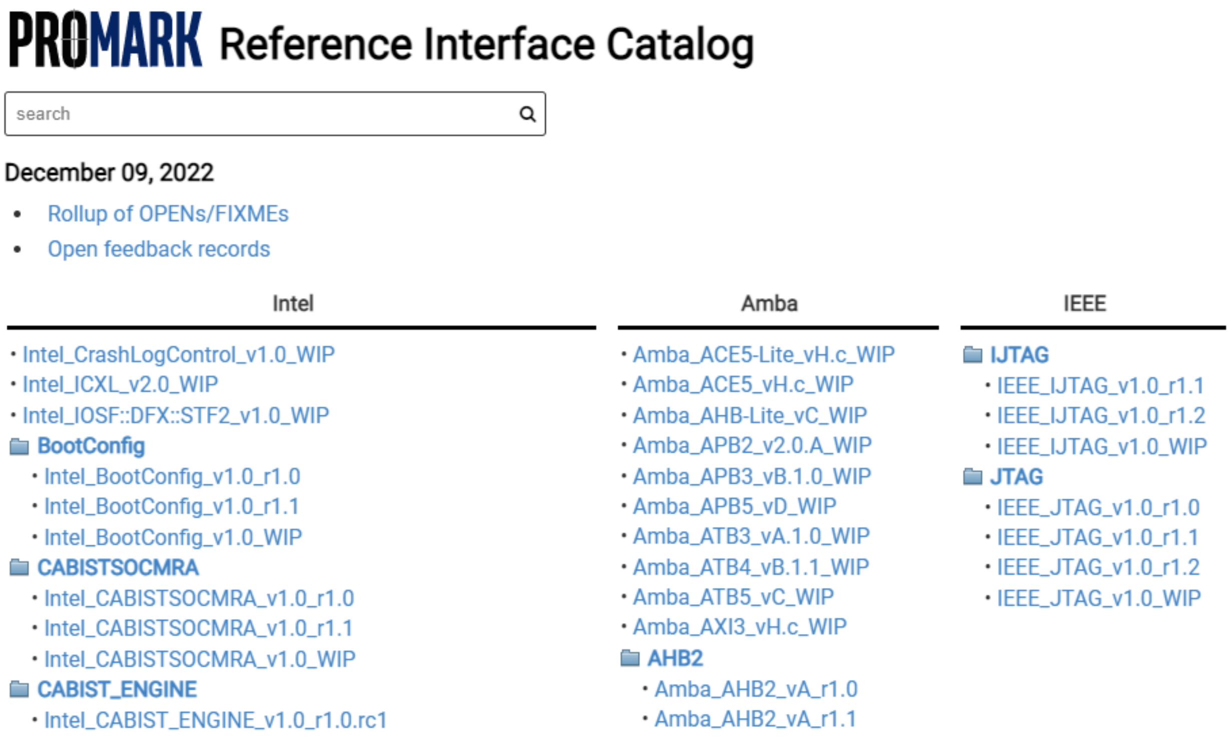
6. ProMark Reference Interface Catalog

The ProMark team maintains a large Reference Interface Catalog for Standard Reference Workflows. This catalog contains a tree of all standard interfaces as Excel files built with MaRS / ACES. These files are available for use in building IP applications and test cases.

Standard Reference Interface Workflow

Here is the catalog page:
https://docs.intel.com/documents/iparch/iptech/Catalog/ReferenceInterfaces/index.html

The catalog page is searchable, and it has a scrollable list of the supported standard interfaces, as shown here:


PROMARK Reference Interface Catalog

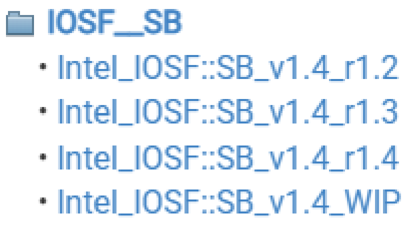
search

December 09, 2022

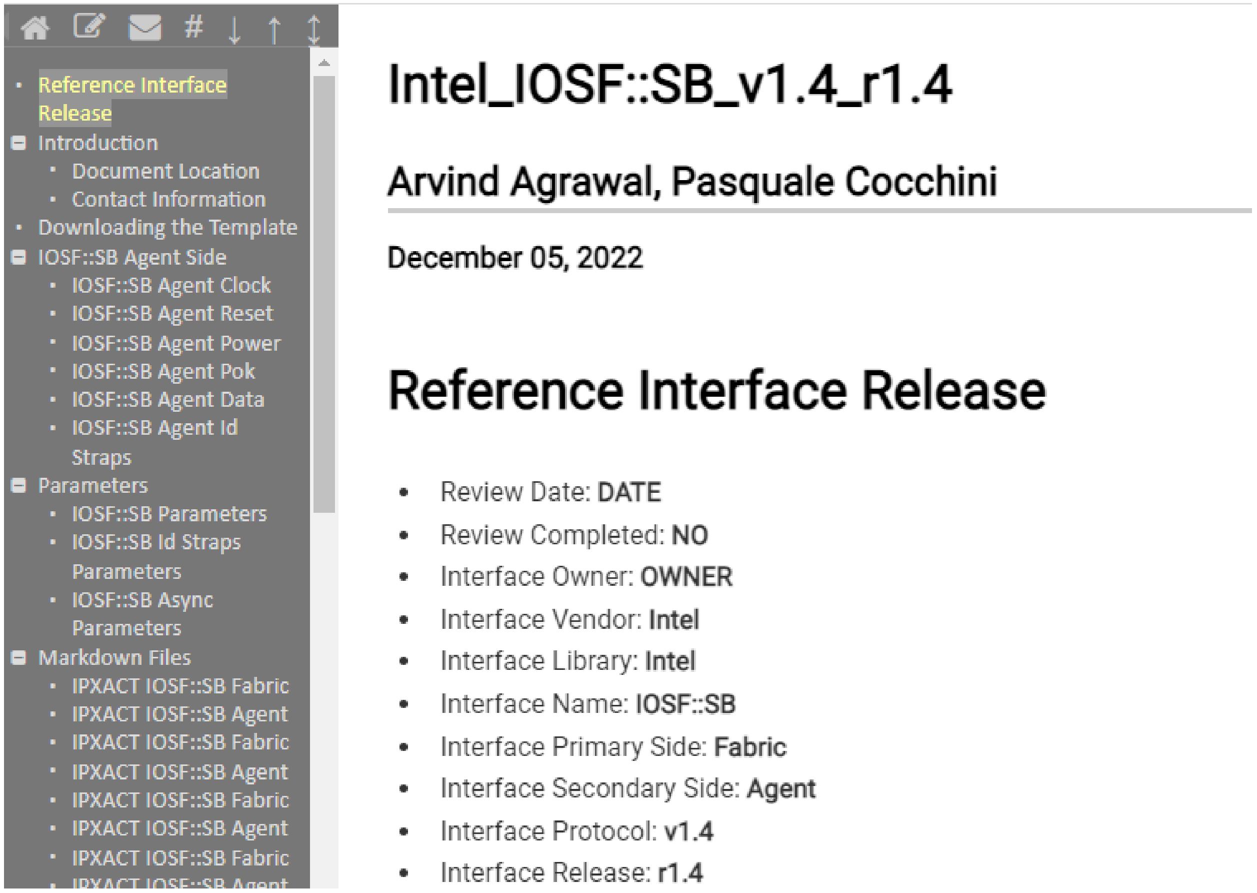
- Rollup of OPENs/FIXMEs
- Open feedback records

Intel	Amba	IEEE
• Intel_CrashLogControl_v1.0_WIP	• Amba_ACE5-Lite_vH.c_WIP	IJTAG
• Intel_ICXL_v2.0_WIP	• Amba_ACE5_vH.c_WIP	• IEEE_IJTAG_v1.0_r1.1
• Intel_IOSF::DFX::STF2_v1.0_WIP	• Amba_AHB-Lite_vC_WIP	• IEEE_IJTAG_v1.0_r1.2
BootConfig	• Amba_APB2_v2.0.A_WIP	• IEEE_IJTAG_v1.0_WIP
• Intel_BootConfig_v1.0_r1.0	• Amba_APB3_vB.1.0_WIP	JTAG
• Intel_BootConfig_v1.0_r1.1	• Amba_APB5_vD_WIP	• IEEE_JTAG_v1.0_r1.0
• Intel_BootConfig_v1.0_WIP	• Amba_ATB3_vA.1.0_WIP	• IEEE_JTAG_v1.0_r1.1
CABISTSOCMRA	• Amba_ATB4_vB.1.1_WIP	• IEEE_JTAG_v1.0_r1.2
• Intel_CABISTSOCMRA_v1.0_r1.0	• Amba_ATB5_vC_WIP	• IEEE_JTAG_v1.0_WIP
• Intel_CABISTSOCMRA_v1.0_r1.1	• Amba_AXI3_vH.c_WIP	
• Intel_CABISTSOCMRA_v1.0_WIP	AHB2	
CABIST_ENGINE	• Amba_AHB2_vA_r1.0	
• Intel_CABIST_ENGINE_v1.0_r1.0.rc1	• Amba_AHB2_vA_r1.1	

The listed download Excel files include a version suffix. The sublists of available files is updated regularly by the development team. You should use the **latest** version, but not the *WIP* version. Scroll the list and select the interface type you are interested in, then left-click on the latest version. In this example, we examine the SideBand interface type:


IOSF__SB
- Intel_IOSF::SB_v1.4_r1.2
- Intel_IOSF::SB_v1.4_r1.3
- Intel_IOSF::SB_v1.4_r1.4
- Intel_IOSF::SB_v1.4_WIP

When the **Intel_IOSF::SB_V1.4_r1.4** is clicked as the latest version, we navigate to the support page for the type:



Intel_IOSF::SB_v1.4_r1.4

Arvind Agrawal, Pasquale Cocchini

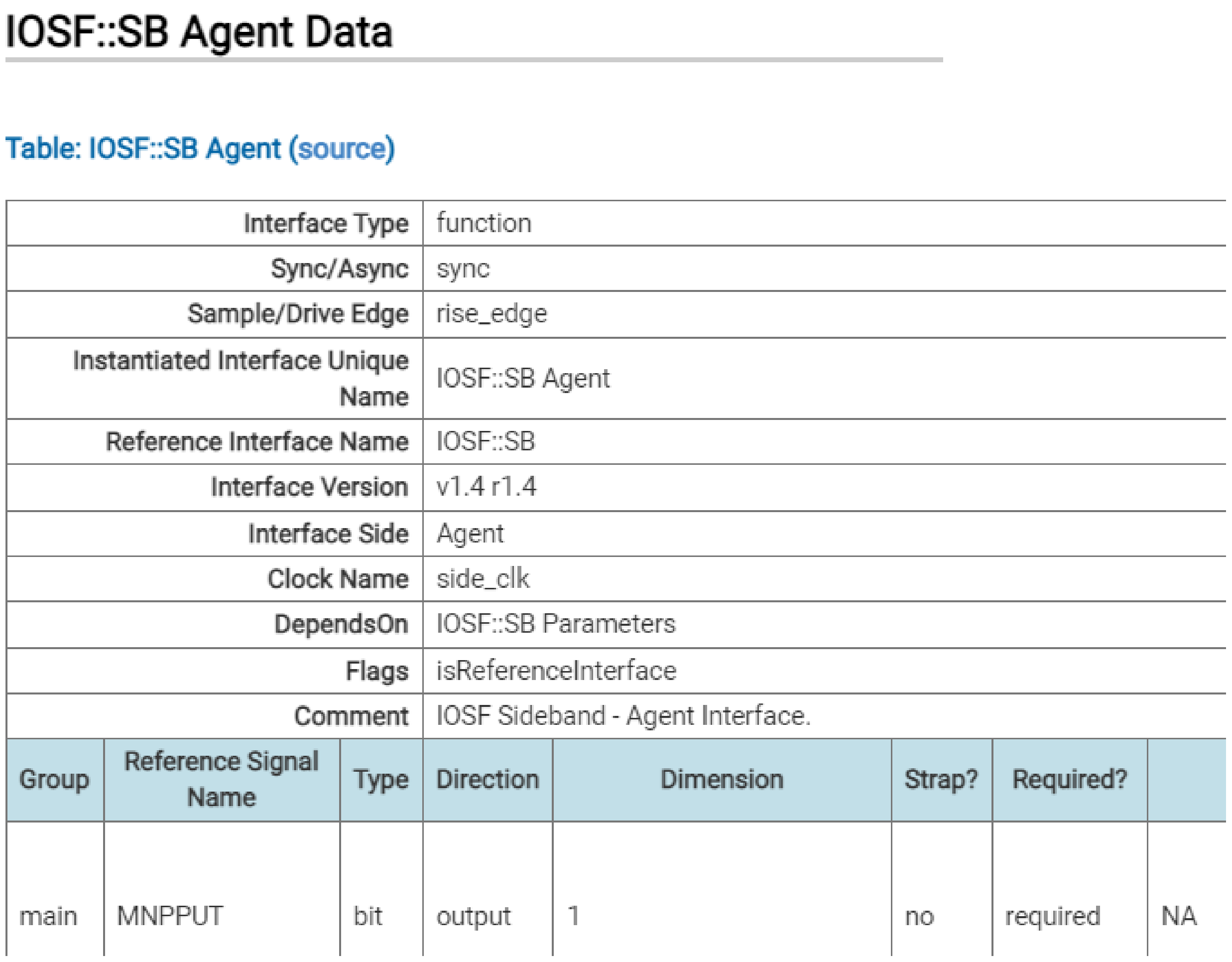
December 05, 2022

Reference Interface Release

- Review Date: **DATE**
- Review Completed: **NO**
- Interface Owner: **OWNER**
- Interface Vendor: **Intel**
- Interface Library: **Intel**
- Interface Name: **IOSF::SB**
- Interface Primary Side: **Fabric**
- Interface Secondary Side: **Agent**
- Interface Protocol: **v1.4**
- Interface Release: **r1.4**

The scrollable support page contains development information and statistics, plus example tables of the supported subtypes. The page also has its own navigation control on the left.

Find a subtype table of interest, such as the **SB Agent Data** example shown here:



IOSF::SB Agent Data

Table: IOSF::SB Agent (source)

Interface Type	function
Sync/Async	sync
Sample/Drive Edge	rise_edge
Instantiated Interface Unique Name	IOSF::SB Agent
Reference Interface Name	IOSF::SB
Interface Version	v1.4 r1.4
Interface Side	Agent
Clock Name	side_clk
DependsOn	IOSF::SB Parameters
Flags	isReferenceInterface
Comment	IOSF Sideband - Agent Interface.

Group	Reference Signal Name	Type	Direction	Dimension	Strap?	Required?	
main	MNPPUT	bit	output	1	no	required	NA

The table's title is clickable, with a link for downloading the Excel **source** data file.

Table: IOSF::SB Agent (source)

Left clicking on the title will send the Excel file to your PC's downloads area. In this example, the file downloaded is **`Intel_IOSF__SB_v1.4_r1.4.xlsx`**. This file is tabbed and preconfigured for use as a standard interface for ACES calls in a ProMark / MaRS application. Move the file to an appropriate place (for example, the **`src`** directory in your local repo tree). You would then use Visual Studio Code to create an **'.mmd'** file with ACES calls to the Excel template using this downloaded file's full name.

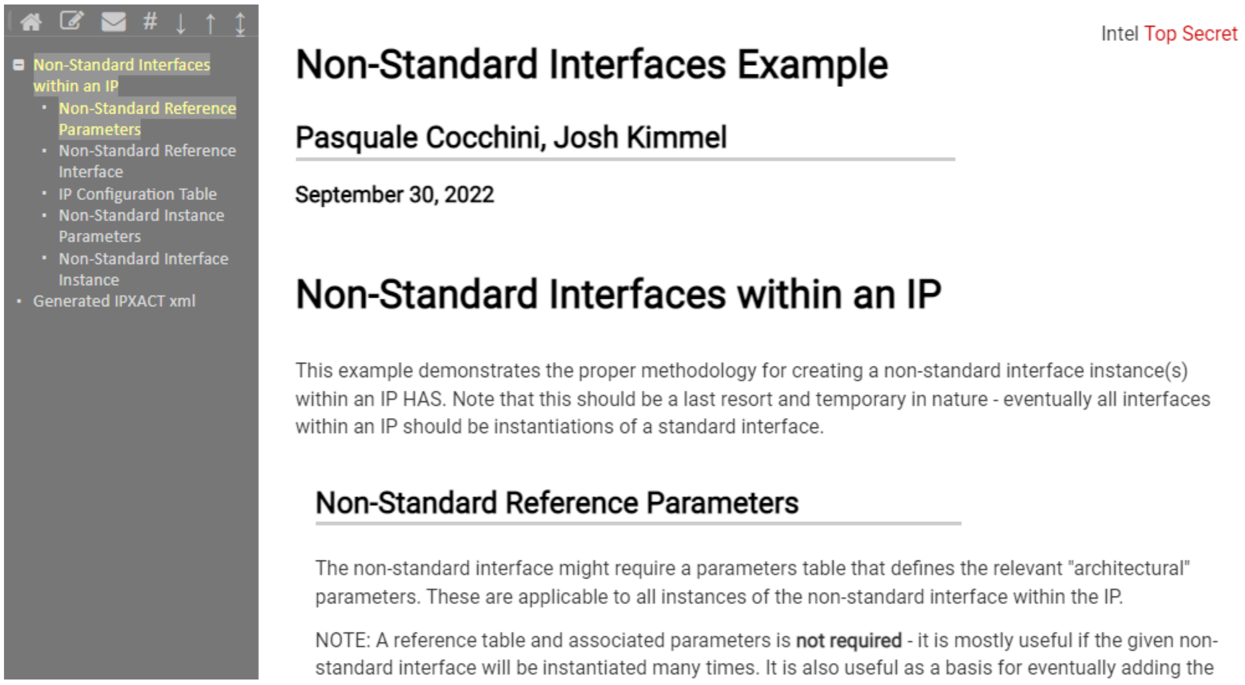
You use these standard interface types to create applications and test cases. An example application is shown accessing a downloaded Excel template in the next section of this guide.

You do **NOT** use these standard interface types to create your own customized variant interfaces.

ProMark / MaRS does provide the means for you to create your own variant interfaces. These interfaces are formally called **'Non-Standard Interfaces'**, and informally they are sometimes called **'Ad Hoc'** interfaces.

Non-Standard Interfaces within an IP

Since these interfaces are inherently non-standard, they don't have a Reference Catalog like standard interfaces. They are supported in ProMark / MaRS, which provides an example page as a resource to help you build applications and test cases. Use this link as a starting point:
https://docs.intel.com/documents/iparch/iptech/Examples/HAS/non_standard.html



Non-Standard Interfaces within an IP
Non-Standard Reference Parameters
Non-Standard Reference Interface
IP Configuration Table
Non-Standard Instance Parameters
Non-Standard Interface Instance
Generated IPXACT xml

Intel Top Secret

Non-Standard Interfaces Example

Pasquale Cocchini, Josh Kimmel

September 30, 2022

Non-Standard Interfaces within an IP

This example demonstrates the proper methodology for creating a non-standard interface instance(s) within an IP HAS. Note that this should be a last resort and temporary in nature - eventually all interfaces within an IP should be instantiations of a standard interface.

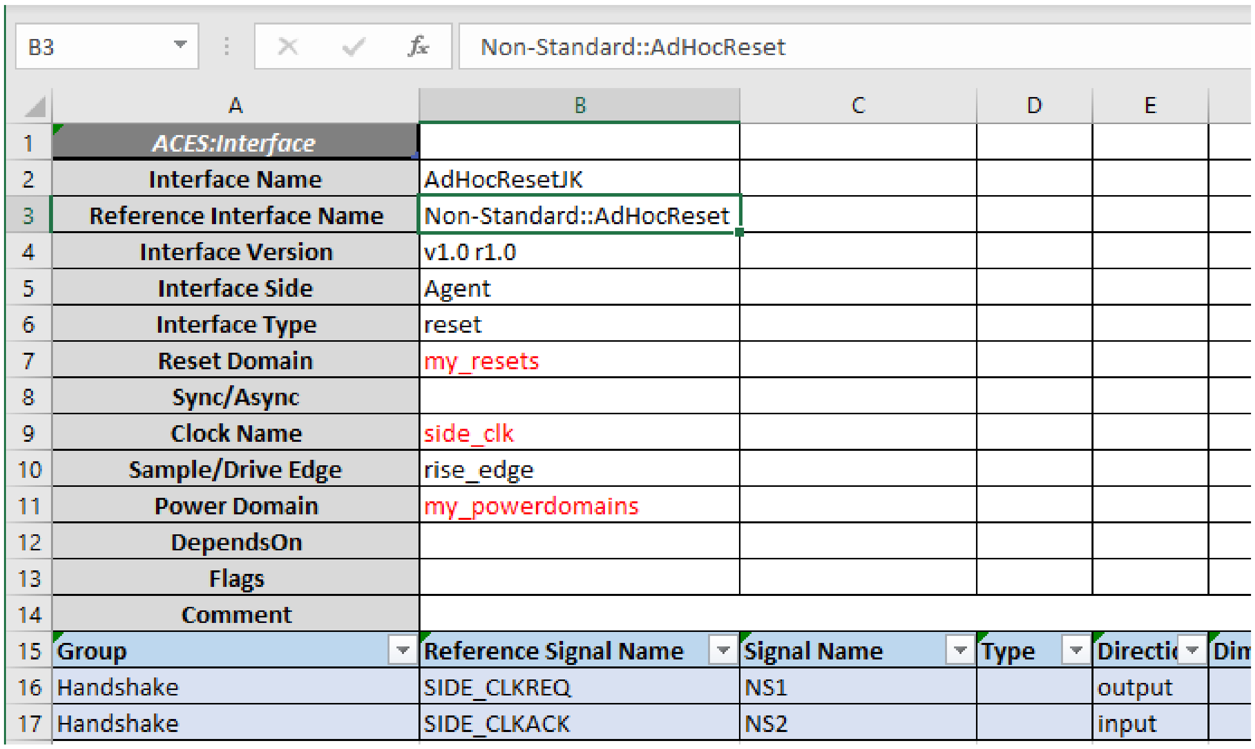
Non-Standard Reference Parameters

The non-standard interface might require a parameters table that defines the relevant "architectural" parameters. These are applicable to all instances of the non-standard interface within the IP.

NOTE: A reference table and associated parameters is **not required** - it is mostly useful if the given non-standard interface will be instantiated many times. It is also useful as a basis for eventually adding the

This help page provides detailed explanations, but it does not have links to Excel files, since these interfaces are **non-standard** by definition. Follow the guidance and examples closely if you need to work with non-standard interfaces in your IP scenario.

When you create a non-standard or ad hoc interface, always start with a downloaded Excel file of the most current standard interface type you are interested in, and then customize that interface file. You MUST designate the Reference Interface Name with the prefix '**Non-Standard**'. In the example image shown below, a ***Reset*** interface is being customized and renamed as to ***AdHocReset*** and the Reference Interface Name has the prefix '**Non-Standard**' so that the field contains the text ***'Non-Standard::AdHocReset'***. This specific detail is necessary for the generated IPXact XML file to express the non-standard interface data. In this example, the fields highlighted in red point to the customization of the interface. The red highlighting is not functional or necessary, but done here for clarity.



B3 | Non-Standard::AdHocReset

	A	B	C	D	E
1	***ACES:Interface***				
2	**Interface Name**	AdHocResetJK			
3	**Reference Interface Name**	Non-Standard::AdHocReset			
4	**Interface Version**	v1.0 r1.0			
5	**Interface Side**	Agent			
6	**Interface Type**	reset			
7	**Reset Domain**	my_resets			
8	**Sync/Async**				
9	**Clock Name**	side_clk			
10	**Sample/Drive Edge**	rise_edge			
11	**Power Domain**	my_powerdomains			
12	**DependsOn**				
13	**Flags**				
14	**Comment**				
15	**Group**	**Reference Signal Name**	**Signal Name**	**Type**	**Directi**	**Dir**
16	Handshake	SIDE_CLKREQ	NS1		output	
17	Handshake	SIDE_CLKACK	NS2		input	

When a ProMark / MaRS application is developed using the IPArch container, a multi-markdown '.mmd' file accesses one or more Excel files with ACES data to compile and build in VSC an HTML file as an image for electronic publication. The development file organization and hierarchy were introduced in Section 4 of this guide.

In addition to the output HTML file, there are other artifacts generated by the build. These produced outputs include intermediate representation files, log files, one or more JSON files, and an XML file structured as an IPXact file. These outputs are the result of Python scripts that run during the build.

The IPXact XML file is one of those script-generated build outputs. This XML file is intended as an intermediate stage in the IP modeling process. This IPXact XML file has signal data and parameters organized and prepared for the next stage of the IP modeling and publication process. For example, a key workflow component that will consume the IPXact file is an Intel application called '***SoC Builder***'.

Generated IPXact xml

There is a link to an IPXact file at the bottom of the non-standard help page, as an example of what this XML file looks like for a non-standard interface. The link is also included here. The generated IPXact file shows the implications for IPXact.
https://docs.intel.com/documents/iparch/iptech/Examples/HAS/mrs/non_standard.ipxact.xml

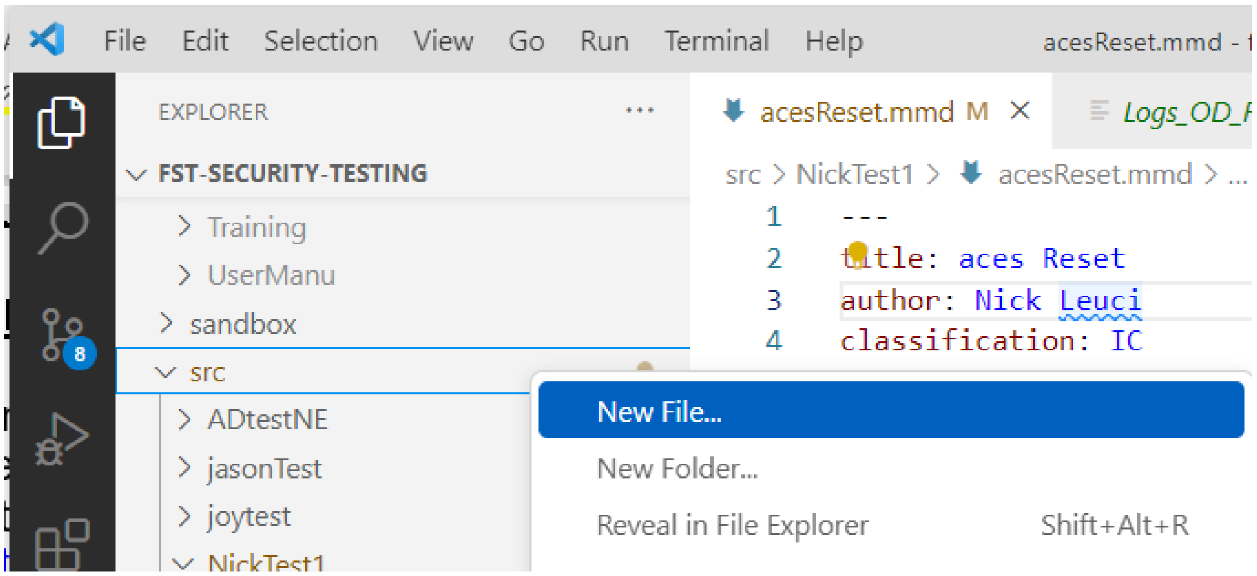
A specific header insert in the '.mmd' file is necessary to properly generate the IPXact XML file. This header is documented in the application example in the next section.

MaRS / ACES is not the only means to generate an IPXact XML file. The Collage tool can also generate this XML file. The SOC Builder tool can consume the IPXact XML file. The IPXact XML file has many aspects and uses within Intel's IP engineering architecture. The following link to an Intel Wiki is a comprehensive guide to those workflow uses:

https://wiki.ith.intel.com/display/IPG/IPG+Architecture+IPXACT+to+Design+Team+IPXACT+Flow#IPGArchitectureIPXACTtoDesignTeamIPXACTFlow-TableofContents

7. Application Example

This part of the guide is a demonstration of creating a simple application putting together everything covered so far. To start, clone or use a test repo to work from (see Section 4). In VSC, create a work area or branch as a test bed. In your repo source **`src`** directory, create a new file with an '.mmd' extension.



```
---
title: aces Reset
author: Nick Leuci
classification: IC
```

This will create a new, empty text file as a multi-markdown application opened in the first (left-most) editing tab of VSC.

Add the '.mmd' header needed to generate the IPXact XML file.

Copy and paste the following text into the very top of your new blank file:

```
---
title: aces Reset
author: Nick Leuci
classification: IC

mars_options:
    mars_xlsx_template_version_match: true
    mars_return_status_mode: shell
    mars_generate_ipxact:
        verify: true
        no-metadata: false
        gen-component: true
        component-name: NickTest1
        component-version: 1.0

symbols:
    PROTOCOL: v1.0
    RELEASE: r1.0
...
```

This header has configuration settings that will be used to generate the IPXact XML file in your build. You can modify the **`title:`** and **`author:`** settings with your data. The detailed explanation of the use of the IPXact header is covered near the bottom of the Standard Reference Interface Workflow, at 'Step 4': https://docs.intel.com/documents/iparch/iptech/Templates/StructuredContent/doc/guides/std_interfaces_workflow.html#the-yaml-header

Save your new '.mmd' file in VSC. It is a good practice to regularly save your edits.

Select the template from the Standard Interface Catalog.

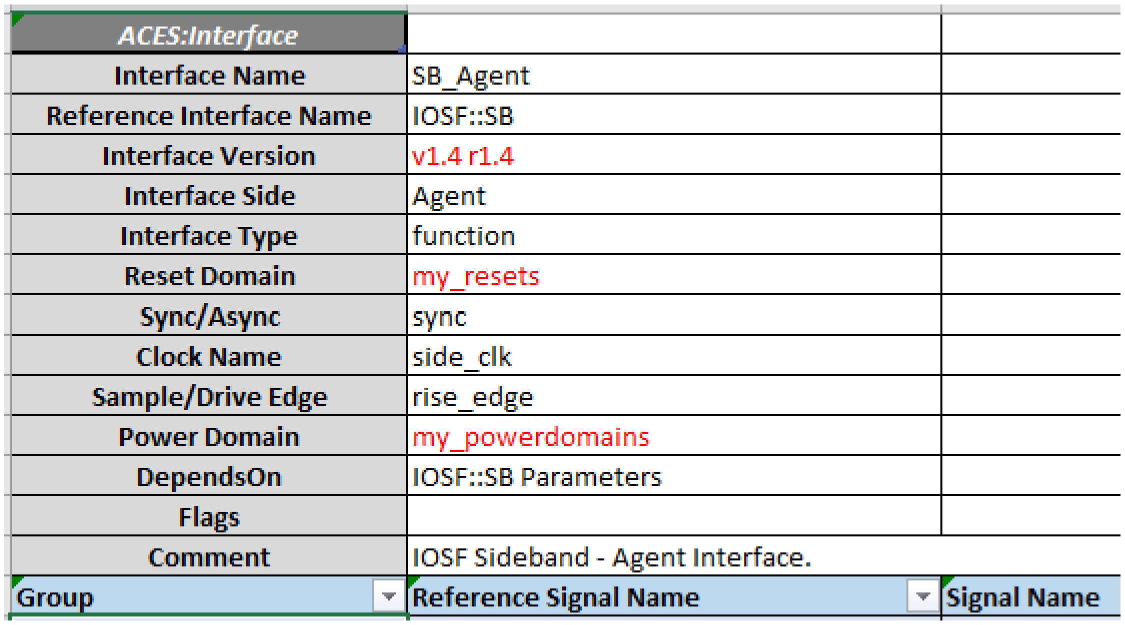
The next step is to use standard interfaces by selecting template sources from the ProMark Standard Interface Catalog (described in Section 6): https://docs.intel.com/documents/iparch/iptech/Catalog/ReferenceInterfaces/index.html

Download the selected Excel template file and move it to the same **`src`** location as your new multi-markdown '.mmd' file. In the following example, the same SideBand standard interface Excel file mentioned previously is used: **Intel_IOSF__SB_v1.4_r1.4.xlsx.**

The standard interface template is the starting point for building your MaRS application. You may need to add signal data, parameters, etc. depending on your purpose. Thus, you will need to do some editing on your Excel template. In the examples shown in this section, the user-customized entries in the Excel template are highlighted in red for clarity. There is no functional need for the text color change.

Add tables, build the application, and resolve all errors.

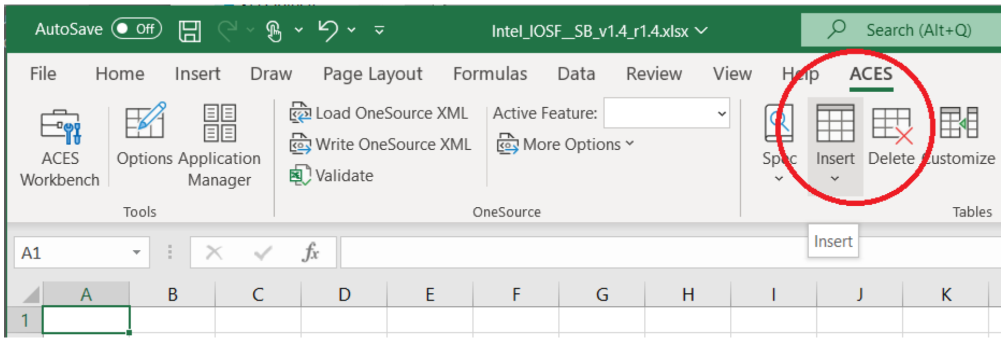
You will get build errors until some important issues are resolved. First, on ALL tabs in the Excel template file, clear the **Flags** field, as shown below:



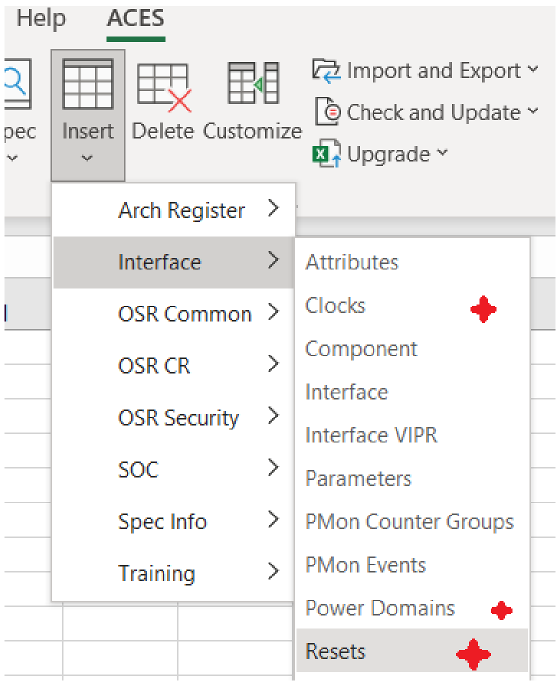
ACES:Interface		
Interface Name	SB_Agent	
Reference Interface Name	IOSF::SB	
Interface Version	v1.4 r1.4	
Interface Side	Agent	
Interface Type	function	
Reset Domain	my_resets	
Sync/Async	sync	
Clock Name	side_clk	
Sample/Drive Edge	rise_edge	
Power Domain	my_powerdomains	
DependsOn	IOSF::SB Parameters	
Flags		
Comment	IOSF Sideband - Agent Interface.	
Group	Reference Signal Name	Signal Name

You may need to tweak the Interface Version. Some of the standard templates have **Reset Domain**, **Clock Name**, and **Power Domain** tables**,** and other don't. If any or all them are not present in your selected template, you will need to add them. Adding one or more of these tables is straightforward. In your Excel template, either click on a new tab, or in an existing tab move the mouse to an open area not resting on an existing table.

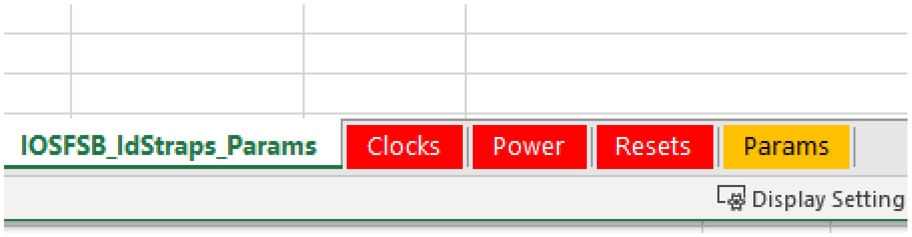
Next, click on the ACES menu control and select the **Insert** icon:



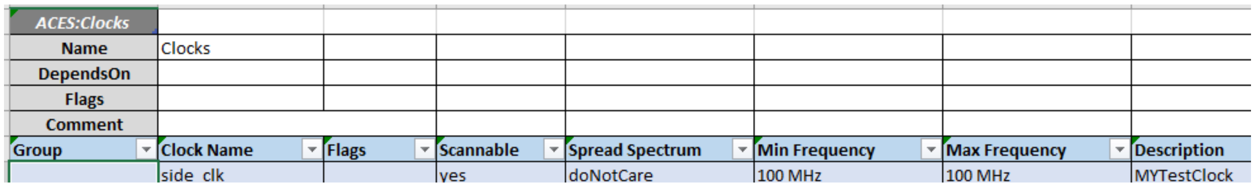
This will raise a floating submenu control allowing you to select the type of object to **Insert**. Select '**Interface'** and another submenu appears. Select the needed type (Clocks, Power Domains, or Resets) and ACES will construct it for you.



For both new or existing tables, you may also need to add data to the **Reset Domain**, **Clock Name**, and **Power Domain** fields. The names in these fields **must match** the values in your ACES application. The names on the Tabs, as shown below in red, are user-friendly but have no functional value.



Click on each of these three tabs / tables and perform the edits that are needed. For example, below are the edits to the **Clocks** tab: Note that the **Clock Name** field matches in both tabs (here **`side_clk`**). That is, the field name in the data (signal) table above **must** match the name in the Clocks table, as shown below.



ACES:Clocks							
Name	Clocks						
DependsOn							
Flags							
Comment							
Group	Clock Name	Flags	Scannable	Spread Spectrum	Min Frequency	Max Frequency	Description
	side_clk		yes	doNotCare	100 MHz	100 MHz	MYTestClock

Depending on your purpose and which standard reference catalog template you selected, how you fill out the signal data and parameter data tables will vary. Here are some guidelines.

For **signal interface schema guidance**, refer to this page:
https://docs.intel.com/documents/iparch/iptech/Templates/StructuredContent/doc/types/ip_interface.html

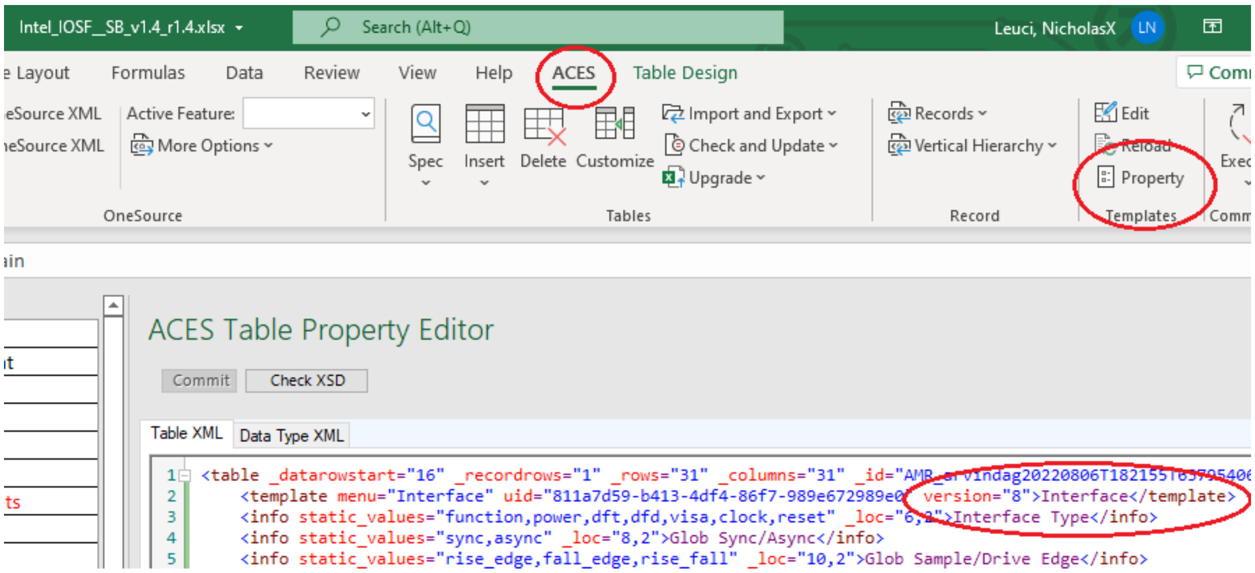
For **parameter schema guidance**, refer to this page:
https://docs.intel.com/documents/iparch/iptech/Templates/StructuredContent/doc/types/ip_parameters.html

Perform your edits and ***SAVE*** your Excel template before doing builds in VSC.

API for ACES Calls

Add your **`aces(     )`** calls in your '.mmd' file, after the header. This API call requires a comma-separated list of arguments. The **first argument** is the **name of the JSON type** that is referenced (**not** the filename). The build script will find the file from the type name in the directory of standard interface types (discussed in Section 5). The name of the **JSON type** for your Excel template can be obtained by doing the following.

1. Open the Excel file to the tab you are going to use and click on the **ACES** menu control at the top of the panel (highlighted in the central small red oval below).
2. The ACES ribbon will display across the top of the panel, allowing you to click on the tab's **Property** (highlighted in the larger red oval below on the right).
3. Use your mouse to drag and expand the **ACES Table Property Editor** to the left, as shown below.
4. The Interface for the tab is shown in the second line of the editor, highlighted in the elongated red oval below.
5. The JSON type for this template is '**`interface`**' and its version is **`'8'`**
6. For this example, you would enter **`interface-v8`** as the first argument of your ACES call in your '.mmd' file.



```
<table _datarowstart="16" _recordrows="1" _rows="31" _columns="31" _id="AMR_arvindag20220806T182155163795406
    <template menu="Interface" uid="811a7d59-b413-4df4-86f7-989e672989e0" version="8">Interface</template>
    <info static_values="function,power,dft,dfd,visa,clock,reset" _loc="6,2">Interface Type</info>
    <info static_values="sync,async" _loc="8,2">Glob Sync/Async</info>
    <info static_values="rise_edge,fall_edge,rise_fall" _loc="10,2">Glob Sample/Drive Edge</info>
```

The **second argument** is the ACES Interface ‘**Interface Name’** field value copied directly from the Excel template file. Compiler errors focused on this second argument refer to it as the ‘Instantiated Interface Name’.

The **third argument** is your created application title for a label in the output HTML, a string prefixed with **`title=`**. This label has no function other than as a visual marker in your constructed HTML page.

The **fourth argument** is the **exact file name** of your Excel template file, a string prefixed by **`acesfname=`**. A path and file name relative to the location of your ‘.mmd’ file is allowed. Any missing files or misspellings will generate compile errors.

Here are two successive ACES calls that use the same Excel interface template:

~~~
```aces("ip_interface-v8", "SB_Agent", title="NICK_6 SB Interface",
acesfname="Intel_IOSF__SB_v1.4_r1.4.xlsx" )
```
~~~

~~~
```aces("ip_parameters-v4", "IOSF::SB Parameters", title="NICK_7 SB
Interface Params", acesfname="Intel_IOSF__SB_v1.4_r1.4.xlsx" )
```
~~~

If you need to add a Clock, Power Domain, or Reset table to your application’s HTML output, use the technique described above to identify the **JSON** type for the table object and use that as your first argument in the ACES call. The table object’s ‘**Name’** field content is the exact text you must use for your second argument. An example of such an ACES call is shown below:

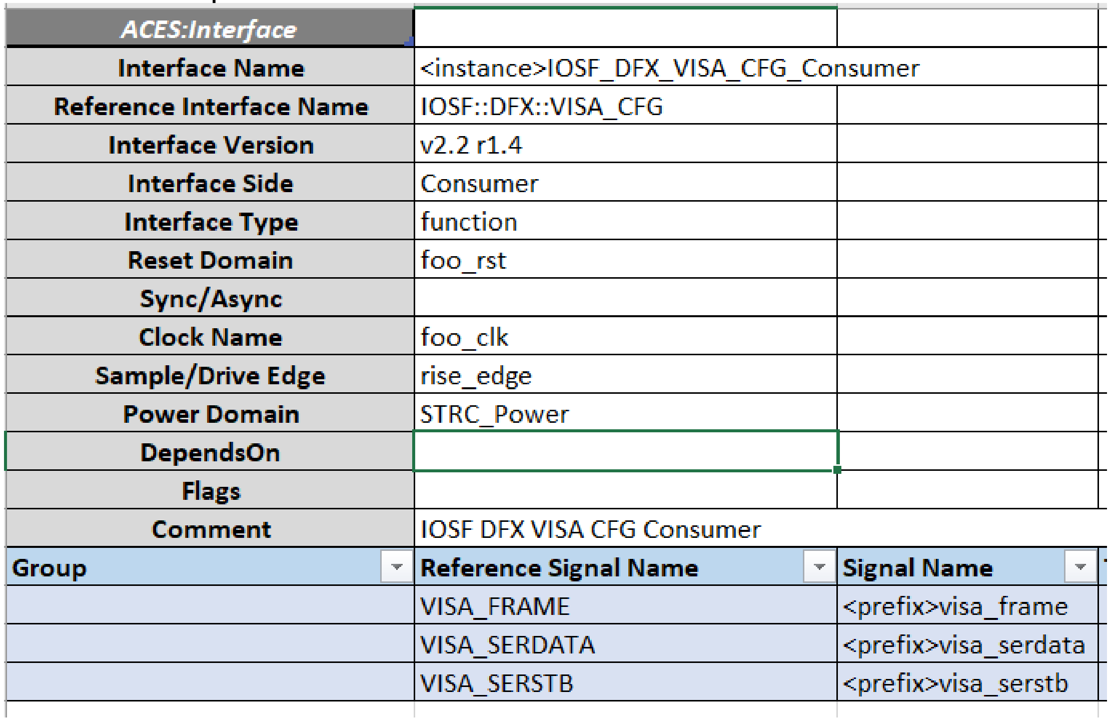
~~~
```aces("ip_clocks-v2", "Clocks", title="NICK_8 SB Clocks Table",
acesfname="Intel_IOSF__SB_v1.4_r1.4.xlsx" )
```
~~~

Note that these ACES calls use **multi-markdown syntax**, and the call's prefix and suffix 'ticks' (` ` `) and the line spacing between **`aces`** API calls are mandatory in your '.mmd' application file. Again, save your '.mmd' file. Also, make certain you have saved any changes made to your Excel template file.

Adding "Variables" to ACES Calls

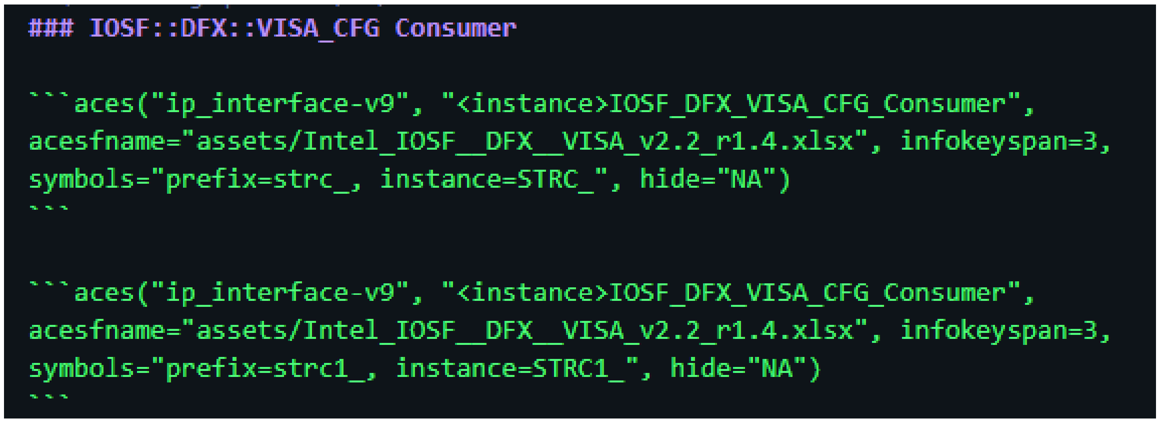
Variables can be added to the Excel ACES table with containing **'<'** and **'>'** characters which can then be given a replaceable value in the '.mmd' ACES call. This macro facility allows using the same ACES table in multiple ACES calls for tasks such as the instantiation of signal names.

Her\e is a sample Excel ACES table with variables:

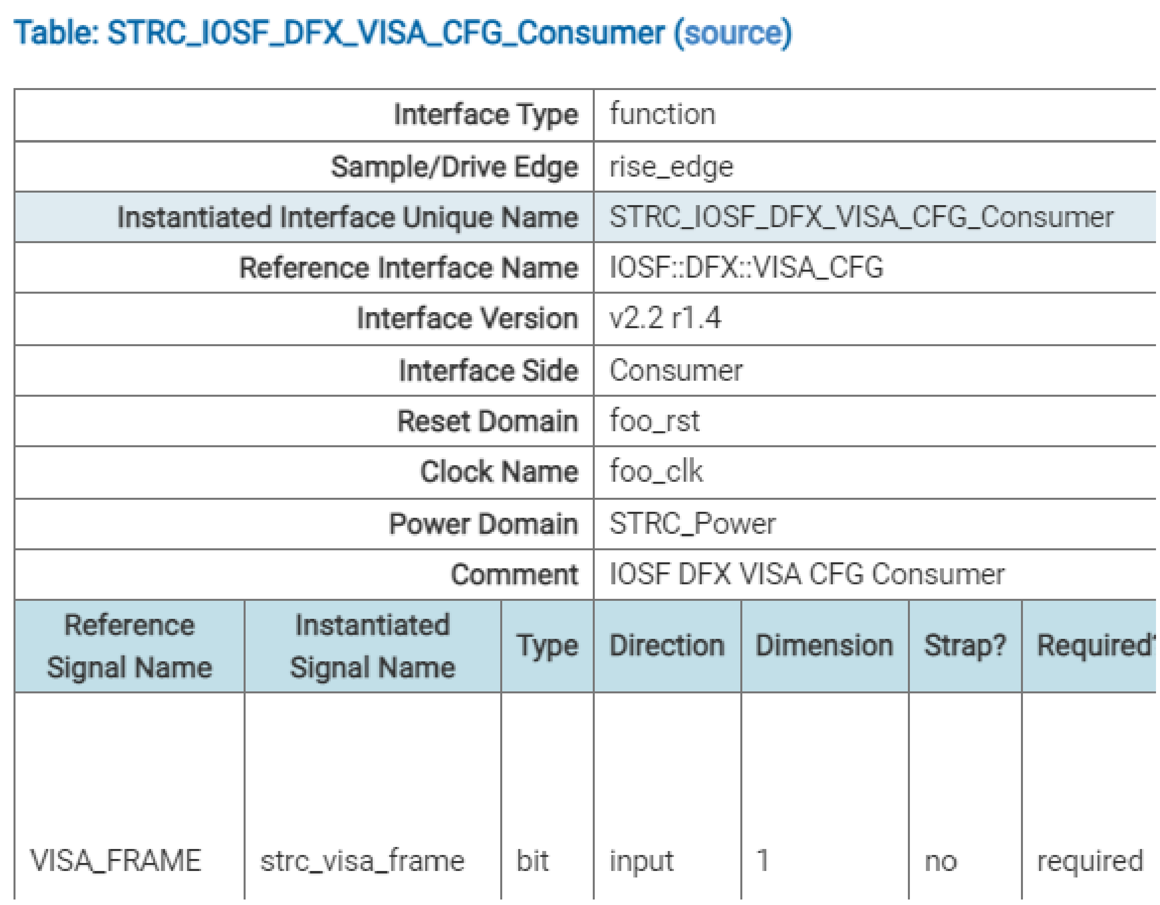


ACES:Interface		
Interface Name	<instance>IOSF_DFX_VISA_CFG_Consumer	
Reference Interface Name	IOSF::DFX::VISA_CFG	
Interface Version	v2.2 r1.4	
Interface Side	Consumer	
Interface Type	function	
Reset Domain	foo_rst	
Sync/Async		
Clock Name	foo_clk	
Sample/Drive Edge	rise_edge	
Power Domain	STRC_Power	
DependsOn		
Flags		
Comment	IOSF DFX VISA CFG Consumer	
Group	**Reference Signal Name**	**Signal Name**
	VISA_FRAME	<prefix>visa_frame
	VISA_SERDATA	<prefix>visa_serdata
	VISA_SERSTB	<prefix>visa_serstb

The values to be inserted in the variable markers in the Excel ACES table are defined in the optional '***symbols'*** parameter of the ACES call. Each call has values for the “prefix” and “instance” variables. Here are two sample ‘.mmd’ ACES calls:



~~~
### IOSF::DFX::VISA_CFG Consumer

```aces("ip_interface-v9", "<instance>IOSF_DFX_VISA_CFG_Consumer",
acesfname="assets/Intel_IOSF__DFX__VISA_v2.2_r1.4.xlsx", infokeyspan=3,
symbols="prefix=strc_, instance=STRC_", hide="NA")
~~~

```aces("ip_interface-v9", "<instance>IOSF_DFX_VISA_CFG_Consumer",
acesfname="assets/Intel_IOSF__DFX__VISA_v2.2_r1.4.xlsx", infokeyspan=3,
symbols="prefix=strc1_, instance=STRC1_", hide="NA")
```
```

Here is the example ‘.mmd’ output from the first ACES call:



Table: STRC_IOSF_DFX_VISA_CFG_Consumer (source)

Interface Type	function
Sample/Drive Edge	rise_edge
Instantiated Interface Unique Name	STRC_IOSF_DFX_VISA_CFG_Consumer
Reference Interface Name	IOSF::DFX::VISA_CFG
Interface Version	v2.2 r1.4
Interface Side	Consumer
Reset Domain	foo_rst
Clock Name	foo_clk
Power Domain	STRC_Power
Comment	IOSF DFX VISA CFG Consumer

Reference Signal Name	Instantiated Signal Name	Type	Direction	Dimension	Strap?	Required
VISA_FRAME	strc_visa_frame	bit	input	1	no	required
```

Here is the example ‘.mmd’ output from the second ACES call:

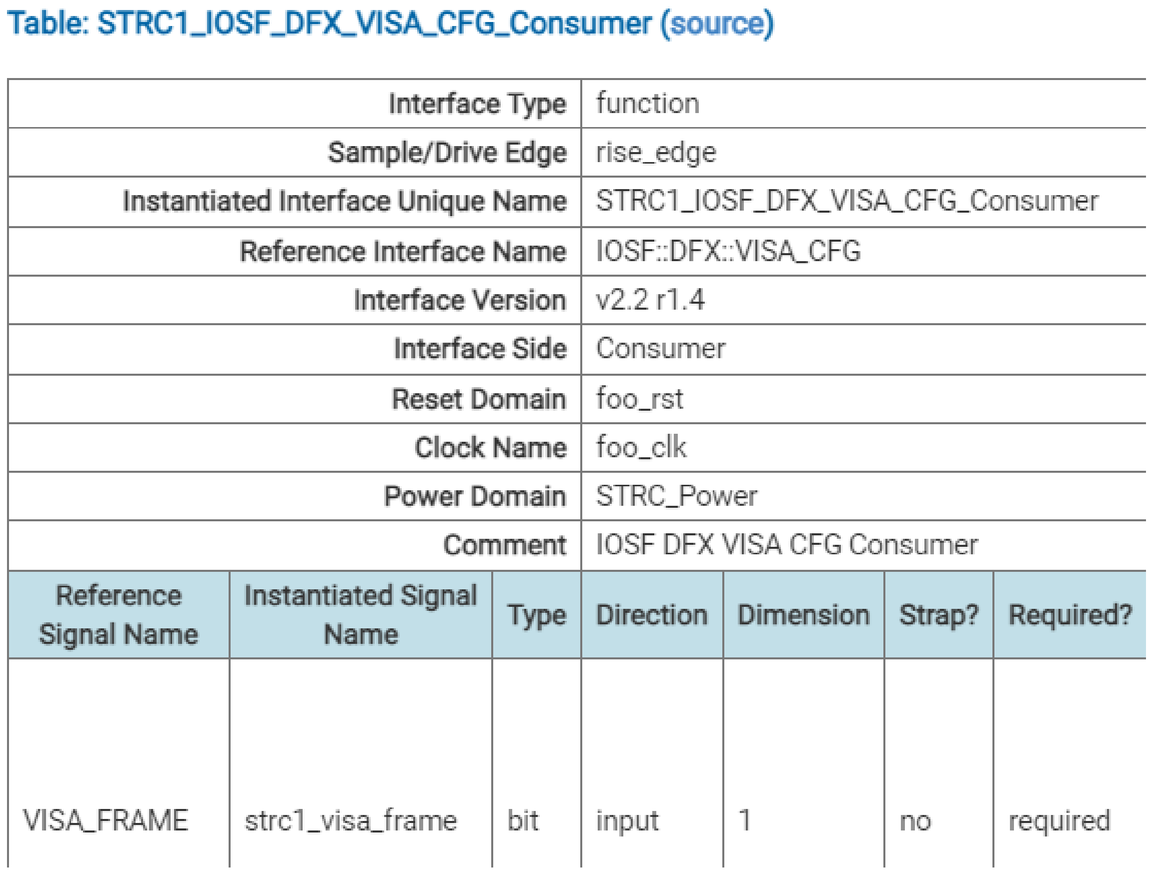


Table: STRC1_IOSF_DFX_VISA_CFG_Consumer (source)

| | |
|---|---|
| Interface Type | function |
| Sample/Drive Edge | rise_edge |
| Instantiated Interface Unique Name | STRC1_IOSF_DFX_VISA_CFG_Consumer |
| Reference Interface Name | IOSF::DFX::VISA_CFG |
| Interface Version | v2.2 r1.4 |
| Interface Side | Consumer |
| Reset Domain | foo_rst |
| Clock Name | foo_clk |
| Power Domain | STRC_Power |
| Comment | IOSF DFX VISA CFG Consumer |

| Reference Signal Name | Instantiated Signal Name | Type | Direction | Dimension | Strap? | Required? |
|---|---|---|---|---|---|---|
| VISA_FRAME | strc1_visa_frame | bit | input | 1 | no | required |

## Construct the HTML output.

At the VSC terminal **$** prompt, enter this build command:

```
$ iparch-mmd2doc yourfilename.mmd --chrome
```

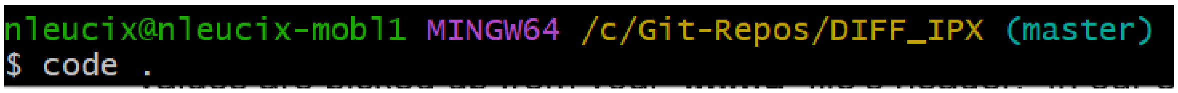
This command will compile your ‘.mmd’ file by calling the Python build scripts to generate the HTML and then launch it in the Chrome browser. If there are compile errors in the VSC terminal area, they will usually give the line number of the object or API call causing the error. The **–chrome** argument is optional but saves a step.

If there are errors, the HTML output will probably not be generated, or it will be generated with error highlights, until the compile errors are fixed. You may also need to edit and fix any errors in the Excel template. When you do get a successful build, your HTML file will be in the **distrib/*yourworkdir*** subtree of your work area and it will have the **base name** of your **‘.mmd’** file with an

‘.html’ extension. This filename is clickable, and it will launch your application in the default browser. There will also be artifact output files such as log files and JSON intermediate files. If you have done a repo-level **`build`**, your **`index.html`** file will also be there.

An Alternative Way to Launch VSC

The narrative here assumes that users will launch Visual Studio Code (VSC) from Windows. Users can also launch VSC directly from Bash. Open the Bash shell and change the location to your desired work area. Type the following command at the terminal’s **$** prompt: **`code .`** (the ‘.’ Is essential).



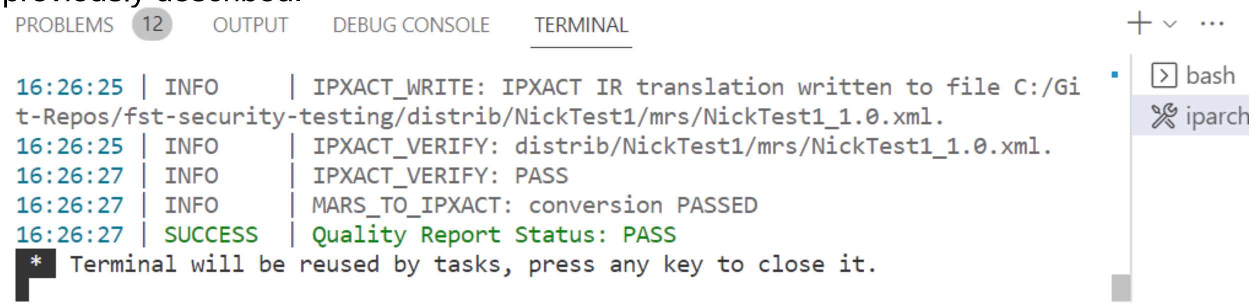
```
nleucix@nleucix-mobl1 MINGW64 /c/Git-Repos/DIFF_IPX (master)
$ code .
```

When you press the **Enter** key, a new instance of Visual Studio Code will be launched at Windows. Use VSC normally to load the repo file location where you wish to work and open your ‘.mmd’ file for editing. Instead of entering build commands at the VSC terminal’s command line, click in your source ‘.mmd’ file (make sure you can see your mouse cursor inside the file). You can now use control command short-cuts to do your builds and display your HTML output.

Here are two such keyboard command macros:

CTRL-m	This does both the equivalent ‘**`iparch-mmd2doc`**’ and ‘**`iparch-build`**’ commands
CTRL-h	This takes the constructed ‘.mmd’ build and launches it in your default browser (the equivalent of ‘**`—chrome`**’)

Note that when you use these keyboard macros, the output messages to the terminal and the build operations are identical to using VSC’s terminal mode as previously described.



```
16:26:25 | INFO     | IPXACT_WRITE: IPXACT IR translation written to file C:/Gi
t-Repos/fst-security-testing/distrib/NickTest1/mrs/NickTest1_1.0.xml.
16:26:25 | INFO     | IPXACT_VERIFY: distrib/NickTest1/mrs/NickTest1_1.0.xml.
16:26:27 | INFO     | IPXACT_VERIFY: PASS
16:26:27 | INFO     | MARS_TO_IPXACT: conversion PASSED
16:26:27 | SUCCESS  | Quality Report Status: PASS
 *  Terminal will be reused by tasks, press any key to close it.
```

These key shortcuts are a handy alternative. You can revert to terminal mode by simply moving the mouse pointer and clicking in VSC’s terminal. You may find it useful to simultaneously have one instance of VSC launched from Windows and another instance of VSC launched from Bash.

Your Generated IPXact XML File

Your generated IPXact XML file is one level lower in the **mrs** subdirectory (**distrib/*yourworkdir*/mrs**). This XML file will have an **'.xml'** extension, and the base name of your work area name and a version number; both of these values are picked up from your **'.mmd'** file's header. In our example these values are:

```
component-name: NickTest1
component-version: 1.0
```

Thus, the IPXact XML file's name in this example would be **NickTest1_1.0.xml**, located in the **distrib/NickTest1/mrs** subdirectory. The IPXact XML file will have your signal (port) data, which would then be used as an input to another application, such as the SOC Builder. An example of this very large file is shown in Appendix A of this document.

Here is a sample **aces** API call using a non-standard interface. The syntax is identical; only the Excel template data is different:

~~~
```aces("ip_interface-v8", "AdHocResetJK", title="NICK_1 NS Reset
Interface", acesfname="Intel_NonStandardInterface.xlsx" )
```
~~~

8. IPXact XML DIFF Checker

Using the IPArch container with ProMark, MaRS, and ACES generates several outputs. The IPXact XML is such an output packed with useful information. This file is an ASCII text file with no embedded elements or control characters. It is an XML file that captures key structured data points for IPArch uses. One of the primary uses of this XML file is as a process input to the SOC **B**uilder and other components of Intel's architecture and development workflows.

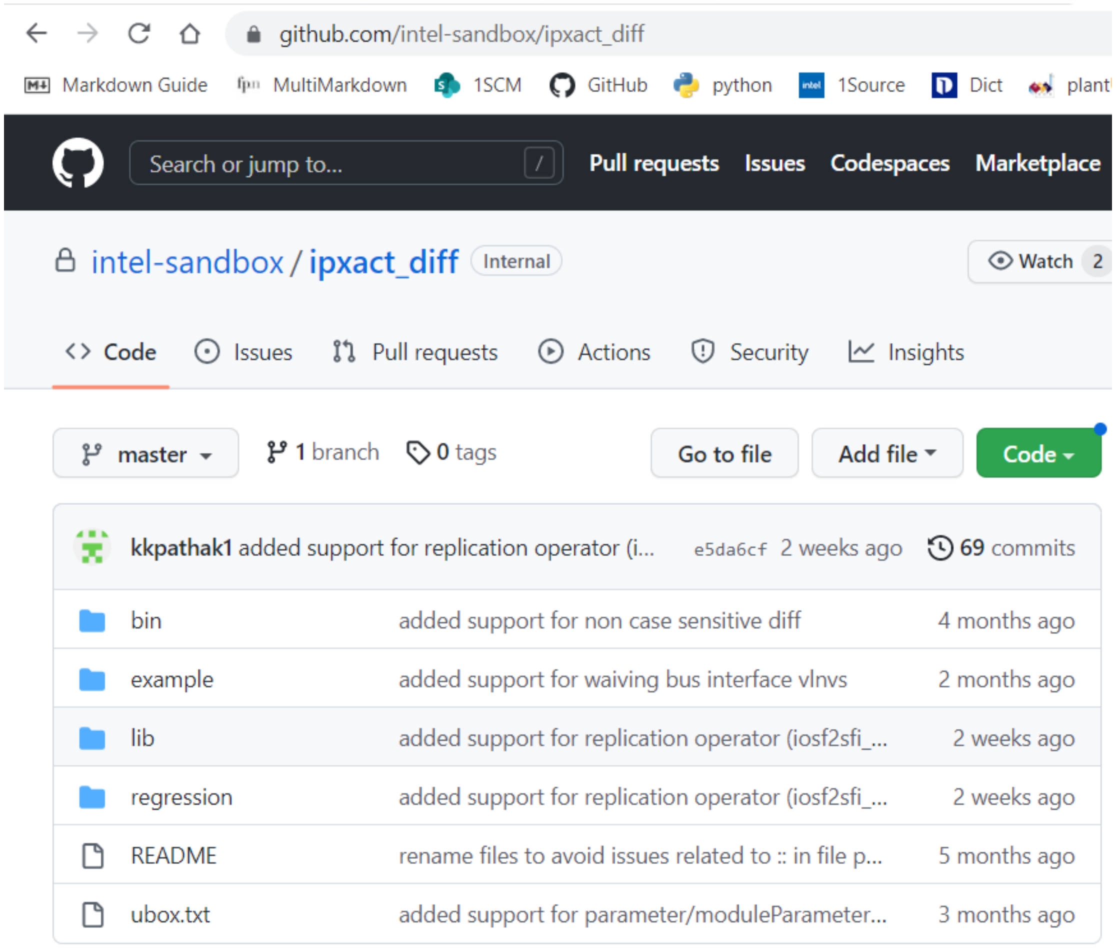
These process workflows are beyond the scope of this guide. However, analyzing and assessing the IPXact XML file's structure and contents is important for IPArch developers. The IPArch engineering group has produced a tool that can be used to assess the IPXact XML file. This tool is the IPXact DIFF checker, a Python module named '**ipxact_diff.py**'.

The IPXact DIFF checker is NOT a general-purpose DIFF analyzer for any text files. It will examine XML files first to see if they are valid IPXact XML files. If the files being checked are detected as not being valid IPXact XML files, the tool will report that and not generate a difference list. If the XML files being checked are valid IPXact XML files, the tool will compare the signals (ports) and attributes of the two files and generate a difference list.

The IPXact DIFF checker is not integrated into the IPArch container environment. It functions independently of the IPArch container, even though it is meant to check the IPArch container's IPXact XML files. The IPXact DIFF checker is available in Intel's GitHub space in its own repo. You can click on this link to clone the repo to your PC (valid account and permissions are required):

https://github.com/intel-sandbox/ipxact_diff.git

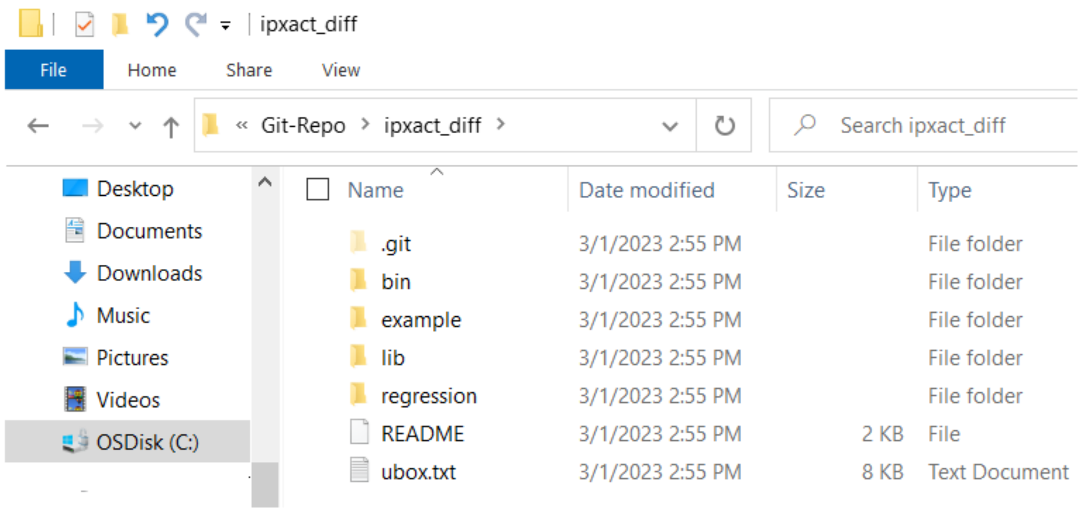
Click on the green '**Code**' button to select the HTTPS clone option.



kkpathak1 added support for replication operator (i... e5da6cf 2 weeks ago 69 commits

bin	added support for non case sensitive diff	4 months ago
example	added support for waiving bus interface vlnvs	2 months ago
lib	added support for replication operator (iosf2sfi_...	2 weeks ago
regression	added support for replication operator (iosf2sfi_...	2 weeks ago
README	rename files to avoid issues related to :: in file p...	5 months ago
ubox.txt	added support for parameter/moduleParameter...	3 months ago

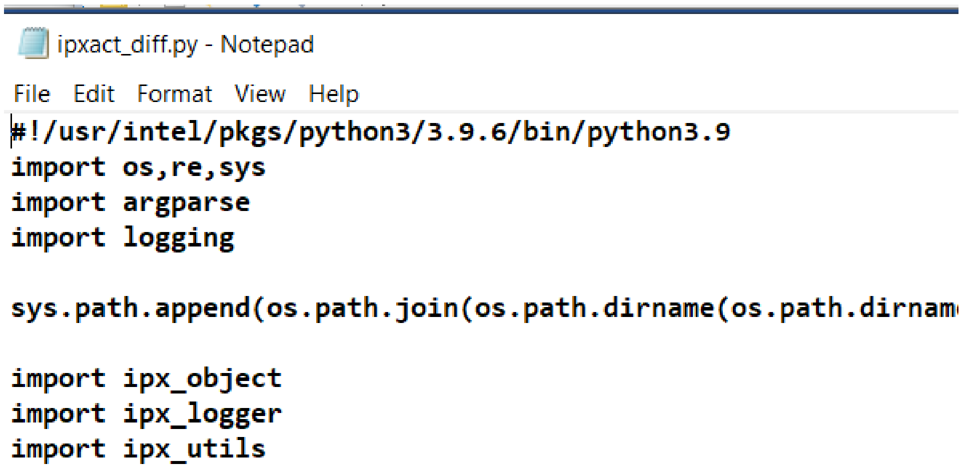
Click the copy icon and then open your BASH shell to the desired location, and at the **$** prompt enter **`git clone`** and paste your copied URL. Git will then clone the repo. Below is an image of the Windows file explorer opened to the cloned repo:



Name	Date modified	Size	Type
.git	3/1/2023 2:55 PM		File folder
bin	3/1/2023 2:55 PM		File folder
example	3/1/2023 2:55 PM		File folder
lib	3/1/2023 2:55 PM		File folder
regression	3/1/2023 2:55 PM		File folder
README	3/1/2023 2:55 PM	2 KB	File
ubox.txt	3/1/2023 2:55 PM	8 KB	Text Document

The **`README`** is a text file that contains input scenarios using the sample IPXact XML data from the **`example`** subdirectory. The **`bin`** directory holds the DIFF checker, a Python file named **`ipxact_diff.py`**. To use the IPXact DIFF checker, you MUST use a BASH shell (terminal) to invoke the Python script with two IPXact XML files as arguments with required keywords.

Here is a partial image of the opened Python file:



```
#!/usr/intel/pkgs/python3/3.9.6/bin/python3.9
import os,re,sys
import argparse
import logging

sys.path.append(os.path.join(os.path.dirname(os.path.dirnam

import ipx_object
import ipx_logger
import ipx_utils
```

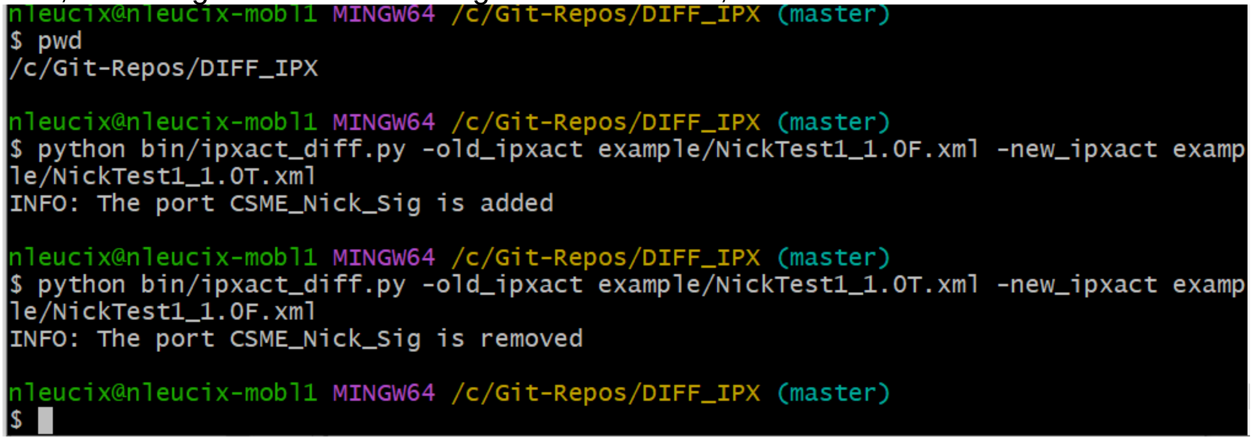
The first line of the file is a location reference to the Intel Engineering Group Server's Python environment, which many users of this guide will NOT be able to access. If you invoke this script directly, you will get a '**Not Found**' error message in your shell. If you have properly installed Python, you already have a valid setup to run this script. Leave the Python script intact. Instead use this substring to invoke the Python script from your Bash shell at the root of the repo: **`python bin/ipxact_dff.py`**. Invoking Python directly overrides the Engineering Group's Server reference in the first line of the script.

Because the IPXact DIFF checker's Python script is not integrated into the Python environment or the IPArch container, it is not on any setup path. You could add the location of the script to your path list in your environment or move the script to somewhere on your path list. Neither of these plausible scenarios are covered here.

You will need to decide for your own needs how best to run the IPXact DIFF checker. For this guide, the assumption is made that the script is run from the root of the repo. This will enable **`python bin/ipxact_dff.py`** to properly be invoked. This Bash command line invocation **requires two arguments** and **allows for additional optional arguments**.

If the argument's required and optional files are not in the same location where the BASH shell is opened, **relative or absolute path names** must be provided with the file names. This makes using the checker a bit awkward, since the text string for a full invocation could be quite lengthy. A practical solution is to build a set of command line full invocations stored in a handy file. This author uses **Notepad++** for that purpose. Another handy resource is your **OneNote** application, standard with Microsoft Office 365. If you are going to use this checker repeatedly, you may want to use the **INTERFACE_DIR option** for a standard location for your IPXact XML files.

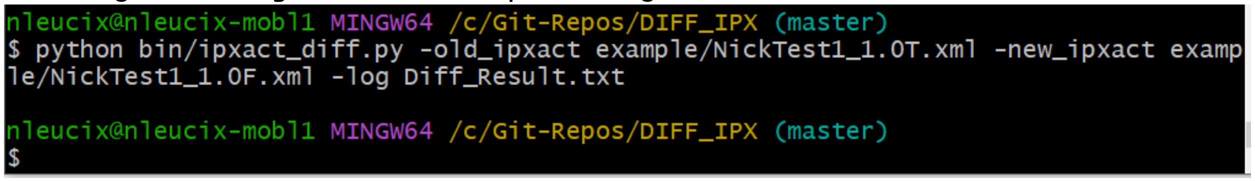
In the screen snap example below, a simple case was constructed to test the presence of the signal 'CSME_Nick_Sig' by changing the state of a dependent parameter from False to True. The output IPXact XML file for each state had an 'F' or 'T' appended to the file name. The DIFF checker was invoked on the two files, switching the order of the argument XML files, as shown here:



```
nleucix@nleucix-mobl1 MINGW64 /c/Git-Repos/DIFF_IPX (master)
$ pwd
/c/Git-Repos/DIFF_IPX

nleucix@nleucix-mobl1 MINGW64 /c/Git-Repos/DIFF_IPX (master)
$ python bin/ipxact_diff.py -old_ipxact example/NickTest1_1.0F.xml -new_ipxact examp
le/NickTest1_1.0T.xml
INFO: The port CSME_Nick_Sig is added

nleucix@nleucix-mobl1 MINGW64 /c/Git-Repos/DIFF_IPX (master)
$ python bin/ipxact_diff.py -old_ipxact example/NickTest1_1.0T.xml -new_ipxact examp
le/NickTest1_1.0F.xml
INFO: The port CSME_Nick_Sig is removed

nleucix@nleucix-mobl1 MINGW64 /c/Git-Repos/DIFF_IPX (master)
$ 
```

The DIFF checker compares the 'new' XML file with the 'old' XML file. In the first invocation, the signal was not in the 'old' file, but it was in the 'new' file, so the signal was reported as '**added'**. In the second invocation, the arguments were reversed, and the checker reported that the signal was in the 'old' (first) file but not in the 'new' (second) file, so the signal was reported as '**removed'**.

In a typical use of the DIFF checker, the comparison list would be long, with dozen or even hundreds of signal (port) changes. The Bash shell would allow scrolling this comparison result list. You can redirect this displayed list to a text file using the **`'-log filename'`** optional argument.



```
nleucix@nleucix-mobl1 MINGW64 /c/Git-Repos/DIFF_IPX (master)
$ python bin/ipxact_diff.py -old_ipxact example/NickTest1_1.0T.xml -new_ipxact examp
le/NickTest1_1.0F.xml -log Diff_Result.txt

nleucix@nleucix-mobl1 MINGW64 /c/Git-Repos/DIFF_IPX (master)
$
```

The comparison list is written to the text file in the current Bash location and not displayed in the shell.

API Reference for the IPXact XML DIFF checker

The IPXact XML DIFF file checker is a robust tool with many possible uses and scenarios. In the interest of completeness and clarity, the developer's API reference for the tool is included here. The following pages contain the developer's Bash command line API reference for using this Python script.

`python bin/ipxact_diff.py -h`

usage: **`ipxact_diff.py [-h] -old_ipxact`** OLD_IPXACT_FILE **`-new_ipxact`** NEW_IPXACT_FILE **`[-interface_dir`** INTERFACE_DIR**`]`** **`[-waiver_file`** WAIVER_FILE**`]`** `[`**`-log`** LOG_FILE**`]`** **`[-debug]`**

optional arguments:

`-h, --help` show this help message and exit

`-old_ipxact` OLD_IPXACT.XML

The old IPXact XML file is a required argument

`-new_ipxact` NEW_IPXACT.XML

The new IPXact file is a required argument

`-interface_dir` INTERFACE_DIR

The optional IPXact standard interface directory

`-waiver_file` WAIVER_FILE

The optional path of the waiver file

`-log` LOG_FILE

The optional log file name (default is STDOUT)

`-debug`

Enable optional debug messages

This is the paradigm for running the IPXact DIFF checker in BASH:

```
$ python bin/ipxact_diff.py -old_ipxact old.xml -new_ipxact new.xml [options]
```

This DIFF utility reports the differences (if any) in old vs new IPXact files e.g., what is being added, removed, and modified.

In addition, the script exit status will be

- non-zero, if there are differences or errors (the exact value will be the number of errors plus the number of differences)
- zero, if there are no differences or errors.

The IPXact DIFF script also provides the capability of waiving of specified interfaces, module parameters, ports etc. through a **waiver file**. In this case it will not report the differences for the specified interface, module parameter and ports etc. The waiver file is a CSV file which will specify the elements which need to be waived off. Such elements are not considered for comparison. As of now following type of elements can be waived off using the specified syntax

- bus interface: "*BUS_INTERFACE*,<name or pattern>"
- bus interface configuration value : "*CONFIGURABLE_ELEMENT_VALUE*, <bus interface name>.<parameter name>"
- bus interface port map : "PORT_MAP, <bus interface name>.<logical port name>"
- port : "*PORT*,<name or pattern>"
- moduleParameter: "*MODULE_PARAMETER*, <name or pattern>"
- parameter: "*PARAMETER*,<name or pattern>"

Note: The pattern can be a **valid regular expression**. For example: **.*** matches anything, **abc.*** matches anything starting with abc etc.

The example syntax of waiver files is in **example/waiver.txt** file of the repo.

Running IPXact diff:

- ipxact_diff on bus definition files
  - python bin/ipxact_diff.py -old_ipxact example/2014/Intel_Intel_IOSF::SB_1.0.xml -new_ipxact example/2014/Intel_Intel_IOSF::SB_1.2.xml
- ipxact_diff on abstraction definition files
  - python bin/ipxact_diff.py -old_ipxact example/2014/Intel_Intel_IOSF::SB_1.0_rtl.xml -new_ipxact example/2014/Intel_Intel_IOSF::SB_1.2_rtl.xml
- ipxact_diff on component files
  - python bin/ipxact_diff.py -old_ipxact example/2014/iocachedmr_arch.xml -new_ipxact example/2014/iocachedmr_design.xml
  - python bin/ipxact_diff.py -old_ipxact example/2014/iocachedmr_arch.xml -new_ipxact example/2014/iocachedmr_design.xml -log diff.txt

    - python bin/ipxact_diff.py -old_ipxact example/2014/iocachedmr_arch.xml -new_ipxact example/2014/iocachedmr_design.xml -waiver_file example/waiver.txt -log diff.txt -interface_dir /nfs/site/disks/crt_linktree_1/rtl_cad/intel/bus_interface_defs/0.13
- ipxact_diff on design files
    - python bin/ipxact_diff.py -old_ipxact example/2009/cxlcm_host_regs_System_design.xml -new_ipxact example/2014/cxlcm_host_regs_System_design.xml

Diff Rules:

1. The detailed list below summarizes what is compared
2. The description and vendor extensions are not compared. For cases where native IPXact elements are described in Vendor extensions, the tool tries to compare such elements. For example, bus interface configurable elements.
3. The isPresent element, if present and evaluates to false, the corresponding element is not compared (bus interfaces, ports)
4. For physical ports where isPresent value evaluates to false, those ports if present in portMaps section of any bus interface, the corresponding portMap is not compared.
5. The port width expressions are evaluated before comparison
6. In some cases, expressions are not possible to evaluate in python (e.g. expression using SV functions $clog2 etc). in such case the expressions are not evaluated and the raw value is being compared
7. Module parameters and parameters where resolve attribute set to "immediate" are not compared.

Diff Waivers

1. Due to coretool limitations, while comparing version string in VLNV, the 'v' is stripped off
2. The vendor/library in VLNV is not compared because in some cases MaRS arch HAS has IEEE as vendor/library whereas coretool always generates "Intel" for vendor/library
    a. Thus waivers can be removed in future because the TCL interface files generated from MaRS HAS includes the capability to add vendor/library as defined in IPXact files

IPXact diff for bus definitions

When comparing two IP-Xact bus definition files, this utility compares the following elements and report out any differences (if any)

- Root vlnv
- The elements isAddressable, directConnection, bradcast, maxMasters, maxSlaves (added/removed/modified)
- The parameters (added/removed/modified)
    - value
    - attributes: type, resolve

IPXact diff for abstraction definitions

When comparing two IP-Xact abstraction definition files, this utility compares the following elements and report out any differences (if any)

- Root vlnv
- The bus Type vlnv

- The logical ports (added/removed)
- The logical ports (modified)
    - The isPresent element
    - The qualifier element
    - The onMaster/onSlave constraints - presence, width, direction (added/removed/modified)
- The parameters(added/removed/modified)
    - value
    - attributes: type, resolve

IPXact diff for component

When comparing two IP-Xact component files, this utility compares the following elements and report out any differences (if any)

- Root vlnv
- The Bus interfaces (added/removed)
- The Bus interfaces (modified)
    - The bus Type vlnv
    - The abstraction type vlnv
    - The logical vs physical port mapping(added/removed/modified)
    - The bus interface parameter configuration(added/removed/modified)
- The top module name(modified)
- The top module parameters(added/removed/modified)
    - value
    - attributes: type, resolve
- The physical port(added/removed)
- The physical port(modified)
    - The direction(modified)
    - The range (added/removed/modified)
- The parameters(added/removed/modified)
    - value
    - attributes: type, resolve

IPXact diff for design

When comparing two IP-Xact design files, this utility compares the following elements and report out any differences (if any)

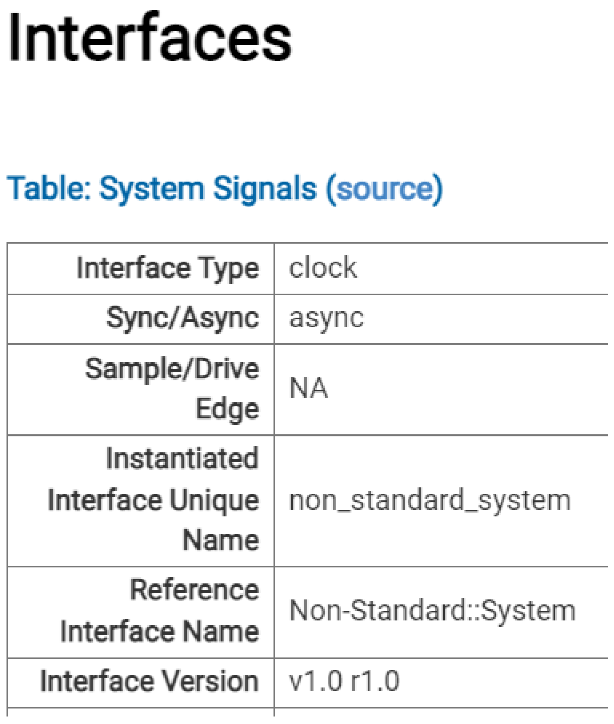
- Root vlnv
- The instances (added/removed)
- The instances (modified)
    - The component reference vlnv (modified)
    - The component module parameter configuration(added/removed/modified)
- Connections
    - To be supported

9. IP HAS Template

If you need to create a framework for a new or customized IP, the IP HAS Template has been created for that purpose. This template serves as a generic starting **system** for creating any kind of interface. With it, you can characterize it as needed with the appropriate settings and data by extending the bare metadata template to encompass your application. It has the minimum configuration all interface templates require, including already defined Clocks, Power, and Resets tables, plus Component and SB Messages tables.

The IP HAS template is not in the Standard Reference Catalog. Instead, it is in its own page: https://docs.intel.com/documents/iparch/iptech/Templates/IPHAS_TEMPLATE/IPHAS_TEMPLATE.html

When you click and open this page, you will find extensive references and resources. You can find the metadata system template by scrolling down the page to the Interfaces section:

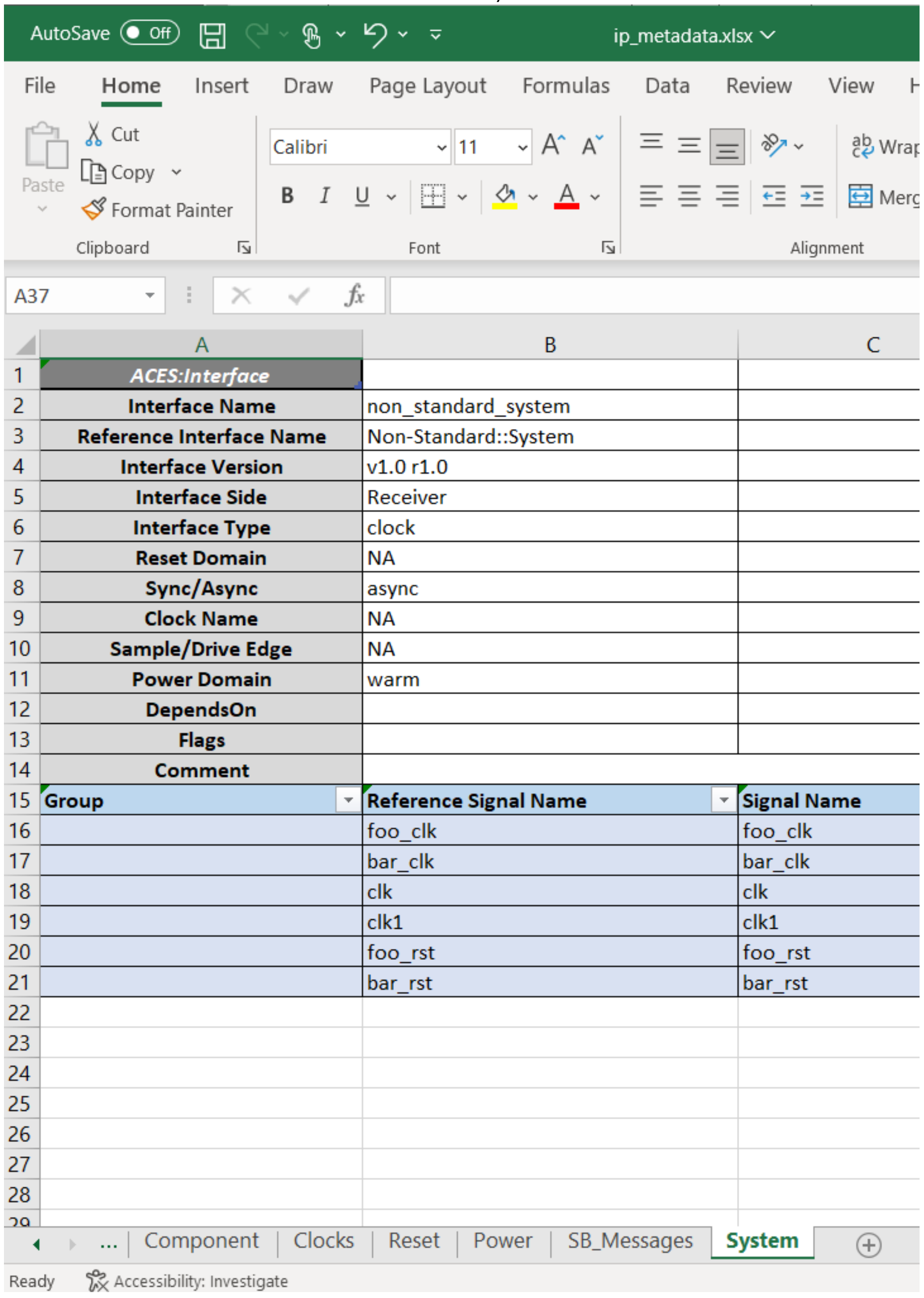


Interfaces

Table: System Signals (source)

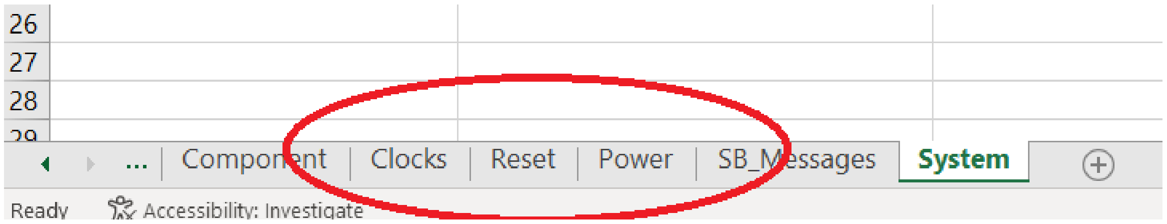
Interface Type	clock
Sync/Async	async
Sample/Drive Edge	NA
Instantiated Interface Unique Name	non_standard_system
Reference Interface Name	Non-Standard::System
Interface Version	v1.0 r1.0

Click on the **(source)** link to download the Excel template: **ip_metadata.xlsx.** Move the template to your designated source work area for building the '.mmd' application. When you open this template in Excel, you will note that several tabs with defined tables have been created, as shown below.

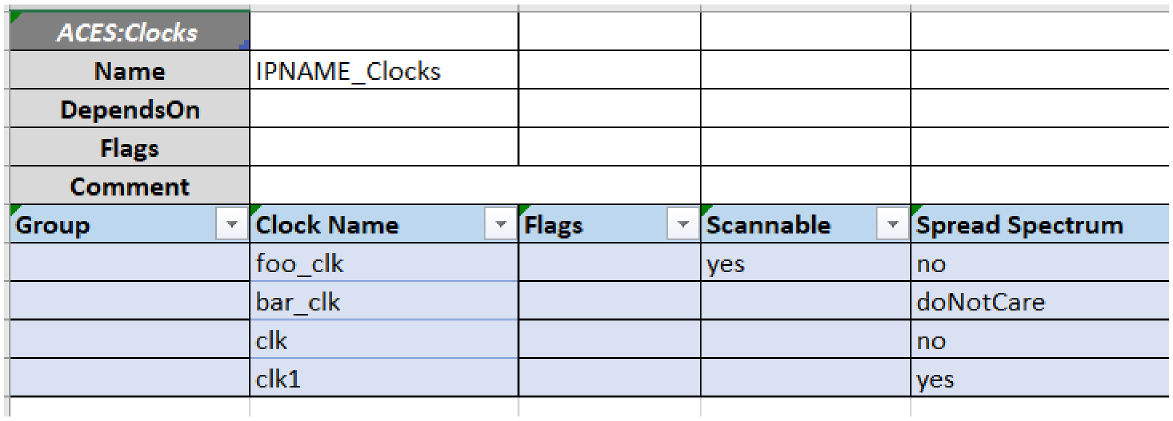

AutoSave Off
ip_metadata.xlsx
File Home Insert Draw Page Layout Formulas Data Review View
Paste Cut Copy Format Painter
Clipboard
Calibri 11
Font
Wrap Merg
Alignment
A37
Component Clocks Reset Power SB_Messages System
Ready Accessibility: Investigate

	A	B	C
1	*ACES:Interface*		
2	**Interface Name**	non_standard_system	
3	**Reference Interface Name**	Non-Standard::System	
4	**Interface Version**	v1.0 r1.0	
5	**Interface Side**	Receiver	
6	**Interface Type**	clock	
7	**Reset Domain**	NA	
8	**Sync/Async**	async	
9	**Clock Name**	NA	
10	**Sample/Drive Edge**	NA	
11	**Power Domain**	warm	
12	**DependsOn**		
13	**Flags**		
14	**Comment**		
15	**Group**	**Reference Signal Name**	**Signal Name**
16		foo_clk	foo_clk
17		bar_clk	bar_clk
18		clk	clk
19		clk1	clk1
20		foo_rst	foo_rst
21		bar_rst	bar_rst

The System tab has some clock signals already defined. You will also see the tabs for the Clocks, Reset and Power tables, as indicated here:



Click on the Clocks tab and you will see four predefined clock signals. You can use them or add other clock signals as needed.



ACES:Clocks				
Name	IPNAME_Clocks			
DependsOn				
Flags				
Comment				
Group	Clock Name	Flags	Scannable	Spread Spectrum
	foo_clk		yes	no
	bar_clk			doNotCare
	clk			no
	clk1			yes

The Power and Resets tables in their respective tabs have the minimum settings needed to build an application. Of course, you can modify or add to these tables for your application.

Here are some ACES calls for a starter '.mmd' application built with the IP HAS Template:

**```aces("ip_interface-v9", "non_standard_system", title="IP HAS Template Interface", acesfname="ip_metadata.xlsx")**
**```**

**```aces("ip_clocks-v2", "IPNAME_Clocks", title="IP HAS Template Clocks", acesfname="ip_metadata.xlsx")**
**```**

**```aces("ip_power_domains-v1", "IPNAME_Power", title="IP HAS Temple Power", acesfname="ip_metadata.xlsx")**
**```**

In all respects, working with this template and building an application is consistent with the methods described in this guide. The only difference is that you are not starting from a repo. You can add your application to an existing or new Git repo.

The IP HAS Template page has other tables for Parameters, SB Agent Data, Straps, Debug, and many other significant HAS features. Please remember that this page is a work-in-progress. It will be periodically revised and updated. Bookmark the page and check it regularly.

10. List of Resources

MS Teams is the de facto standard for interactive communication among Intel users and developers of ProMark and MaRS. Here are some Teams channels you can subscribe to get more information and development support:

- FST IP HAS Development and Delivery Team Sync-up
- MARS Initiatives for CSME IP Family
- MaRS/ProMark/OneSource/OsXML
- MARS Troubleshooting

Pasquale Cocchini's team hosts a weekly Teams meeting for MaRS/ProMark Office Hours support sessions, scheduled every **Wednesday at 8:00AM PST** (or PDT during the daylight saving period). This meeting's calendar dates and times are subject to change. Another expert resource is Josh Kimmel.

A good starting point is https://docs.intel.com/documents/il-specs/mars/training/classes/sessions/MaRS%20Training%2022WW15.html. The video recording for the class can be found at https://videoportal.intel.com/channel/MARS++Machine+Readable+Specification/4852.

MARKDOWN / VISUAL STUDIO CODE (VSC)

The **Markdown Guide** is an easy-to-use reference for Markdown that is vendor and browser neutral: see https://www.markdownguide.org/.

Consult this site for a browser and vendor neutral introduction to **Multi-Markdown**: https://fletcherpenney.net/multimarkdown/

ProMark / MaRS supports and recommends VSC, an open-source freely available markdown editor / terminal product. Here is the public link Microsoft provides for downloading VSC: https://code.visualstudio.com/

GITHUB

GitHub access is required, including an Intel GitHub account. You can use this link to commence GitHub setup: GitHub onboarding process

Your GitHub account status is shown in your Intel 1 Source page

Git is organized as an open-source foundation with software, documentation, and related resources. GitHub has both open-source and licensed resources. The freeware GitHub Desktop is a software tool available directly.

Microsoft supports Git and GitHub integration with their Visual Studio.Net product line. Visual Studio.Net is separate and independent of Visual Studio Code (VSC). The Express versions of Visual Studio.Net are free downloads, but Visual Studio.Net is a licensed retail product. However, the Git and GitHub extensions for Visual Studio.Net are free downloads. Check Microsoft's website for further information. **ProMark uses Visual Studio Code, not Visual Studio.Net**.

PROMARK

ProMark has a standard Reference Catalog of interfaces, which are maintained in Excel files.

To install or update ProMark, use this Intel link:
https://docs.intel.com/documents/promark/GitHub/Github_pm_tools.html

The most up-to-date information on ProMark itself can be found at this main site: http://goto/ProMark.

There is another page that is the hub of ProMark tools and methodology uses. You can find this page at: https://docs.intel.com/documents/ProMark/index.html

There is an internal site dedicated to ProMark MaRS training, and this is the best place to start using MaRS:
https://docs.intel.com/documents/il-specs/MaRS/training/classes/index.html

PYTHON

As indicated earlier, use the ProMark installation guide to install and verify Python. Python is a 4th generation open-source language with an enormous base of installed applications and a world-wide group of users. It has many extensions, customizations, and available libraries. For more information on Python, start with the formal organization chartered to standardize and support it: https://www.python.org/

ACES

ACES is developed and owned by a different group other than those responsible for ProMark / MaRS, which are users of ACES. The available documentation for ACES itself is sparse and surprisingly undetailed, given its power and breadth. The ACES release site does contain text documents covering the installation basics. READ THIS CONTENT FULLY. When installing ACES, you need to close ALL Excel files and select the most recent release version of ACES. Download everything and install from your local copy instead of running the

install remotely. If you have installed a previous version of ACES, first remove it completely (via the **Control Panel:: Windows:: Programs** facility). Use this link to navigate to the Intel ACES release site:
\\amr.corp.intel.com\ec\proj\DPGEC\tmg\dts\Tools\DTS CADRoot Releases\ACES\

IPARCH CONTAINER

Detailed instructions for setting up the IPArch container and related tools are presented in this guide, which you should bookmark:
https://docs.intel.com/documents/iparch/iptech/Documentation/ToolSetup.html

For further IPArch information, this link is a good place to take the next step:
https://docs.intel.com/documents/iparch/iptech/Documentation/IPArchContainer.html

Also, here is the link for the ProMark reference page on IPArch and iptech templates:
https://docs.intel.com/documents/iparch/iptech/Templates/index.html

The following link is another, more comprehensive view of Intel's electronic publishing technology. It has reference links to all phases of the process
https://docs.intel.com/documents/iparch/iptech/Examples/HAS/interface_instantiation_with_xlsx_tables.html

IPArch document access permission request guide

Please see below table for the list of common properties for the IPArch Container:

List of IPArch Repos	IPArch/*
Request View Access	[Preferred] AMR\IPArch - Document read BB AMR\IPArch - Document read GB [Old] AMR\ProductEngineering_PEG_FuncDesign_1274_10nm AMR\ProductEngineering_PEG_FuncDesign_Current_IRS AMR\ProductEngineering_PEG_FuncDesign_Future_ITS AMR\Promark - IP Architecture Group HAS GB

Request Publish Access	[Preferred] AMR\IPArch - Document write [Old] AMR\CICG IP Architecture AMR\IPArch_doc_write
Main Owner/Request Access	Ryan Holmqvist
GitHub Repo Access	Read: AGS "iparch all read" Write: AGS "github iparch maintain"
Size	<1.1GB
Release	Manual
DDP Location	https://docs.intel.com/documents/iparch/

IPXACT XML

The IPXact XML file has many aspects and uses within Intel's IP engineering architecture. The following link to an Intel Wiki is a comprehensive guide to those workflow uses:

https://wiki.ith.intel.com/display/IPG/IPG+Architecture+IPXACT+to+Design+Team+IPXACT+Flow#IPGArchitectureIPXACTtoDesignTeamIPXACTFlow-TableofContents

STANDARD REFERENCE INTERFACE

The Standard Reference Interface Workflow is described in detail at this page:
https://docs.intel.com/documents/iparch/iptech/Templates/StructuredContent/doc/guides/std_interfaces_workflow.html#the-yaml-header

Use this link as a starting point for Non-Standard Interfaces:
https://docs.intel.com/documents/iparch/iptech/Examples/HAS/non_standard.html

It is important to note that ProMark allows but does NOT support non-standard interfaces. If you develop and use a non-standard interface, you can request that it be added to Intel's standard interface catalog.

Use this link to access the IP HAS Template:
https://docs.intel.com/documents/iparch/iptech/Templates/IPHAS_TEMPLATE/IPHAS_TEMPLATE.html

INTEL IT SUPPORT

If necessary, contact Intel IT Support for assistance, using these resources:

http://it.intel.com

503-696-1234 (TAC USA hotline)

You will need to properly identify yourself as a Blue Badge or Green Badge employee.

Appendix A.   IPXact XML File Sample

Here is an example of the IPXact XML file generated during a ProMark / MaRS build, found in the **`distrib/`*`localworkdir`*`/mrs`** directory of your local repo:

```
<?xml version="1.0" encoding="utf-8"?>
<ipxact:component xmlns:ipxact="http://www.accellera.org/XMLSchema/IPXACT/1685-2014"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:snps="http://www.synopsys.com/SPIRIT-snps"
xmlns:intel="http://www.intel.com/XMLSchema"
xsi:schemaLocation="http://www.accellera.org/XMLSchema/IPXACT/1685-2014
http://www.accellera.org/XMLSchema/IPXACT/1685-2014/index.xsd">
	<ipxact:vendor>Intel</ipxact:vendor>
	<ipxact:library>Intel</ipxact:library>
	<ipxact:name>NickTest1</ipxact:name>
	<ipxact:version>1.0</ipxact:version>
	<!-- MaRS: Automatically generated with MaRS from architectural specification -->
	<!-- Date: Friday Jan 06 2023 at 14:26:42 Pacific Standard Time -->
	<!-- Node: nleucix-mobl1 on Windows -->
	<ipxact:busInterfaces>
		<ipxact:busInterface>
			<ipxact:name>SB_Agent</ipxact:name>
			<ipxact:description>IOSF Sideband - Agent
Interface.</ipxact:description>
			<ipxact:busType vendor="Intel" library="Intel" name="IOSF::SB"
version="v1.4_r1.4"></ipxact:busType>
			<ipxact:abstractionTypes>
				<ipxact:abstractionType>
					<ipxact:abstractionRef vendor="Intel"
library="Intel" name="IOSF::SB_rtl" version="v1.4_r1.4">
						<ipxact:configurableElementValues>
							<ipxact:configurableElementValue
referenceId="MESSAGEPAYLOADWIDTH">8</ipxact:configurableElementValue>
							<ipxact:configurableElementValue
referenceId="SB_PARITY_REQUIRED">1</ipxact:configurableElementValue>
						</ipxact:configurableElementValues>
					</ipxact:abstractionRef>
					<ipxact:portMaps>
						<ipxact:portMap>
							<ipxact:logicalPort>

	<ipxact:name>MNPPUT</ipxact:name>
							</ipxact:logicalPort>
							<ipxact:physicalPort>

	<ipxact:name>csmea_cse_sb_MNPPUT</ipxact:name>
							</ipxact:physicalPort>
						</ipxact:portMap>
						<ipxact:portMap>
							<ipxact:logicalPort>

	<ipxact:name>MPCPUT</ipxact:name>
							</ipxact:logicalPort>
							<ipxact:physicalPort>

	<ipxact:name>csmea_cse_sb_MPCPUT</ipxact:name>
							</ipxact:physicalPort>
						</ipxact:portMap>
						<ipxact:portMap>
							<ipxact:logicalPort>

	<ipxact:name>MNPCUP</ipxact:name>
							</ipxact:logicalPort>
```

```
                                                        <ipxact:physicalPort>

<ipxact:name>csmea_cse_sb_MNPCUP</ipxact:name>
                                                        </ipxact:physicalPort>
                                                </ipxact:portMap>
                                                <ipxact:portMap>
                                                        <ipxact:logicalPort>

<ipxact:name>MPCCUP</ipxact:name>
                                                        </ipxact:logicalPort>
                                                        <ipxact:physicalPort>

<ipxact:name>csmea_cse_sb_MPCCUP</ipxact:name>
                                                        </ipxact:physicalPort>
                                                </ipxact:portMap>
                                                <ipxact:portMap>
                                                        <ipxact:logicalPort>

<ipxact:name>MEOM</ipxact:name>
                                                        </ipxact:logicalPort>
                                                        <ipxact:physicalPort>

<ipxact:name>csmea_cse_sb_MEOM</ipxact:name>
                                                        </ipxact:physicalPort>
                                                </ipxact:portMap>
                                                <ipxact:portMap>
                                                        <ipxact:logicalPort>

<ipxact:name>MPAYLOAD</ipxact:name>
                                                        </ipxact:logicalPort>
                                                        <ipxact:physicalPort>

<ipxact:name>csmea_cse_sb_MPAYLOAD</ipxact:name>
                                                        </ipxact:physicalPort>
                                                </ipxact:portMap>
                                                <ipxact:portMap>
                                                        <ipxact:logicalPort>

<ipxact:name>MPARITY</ipxact:name>
                                                        </ipxact:logicalPort>
                                                        <ipxact:physicalPort>

<ipxact:name>csmea_cse_sb_MPARITY_NOTSUPPORTED</ipxact:name>
                                                        </ipxact:physicalPort>
                                                </ipxact:portMap>
                                                <ipxact:portMap>
                                                        <ipxact:logicalPort>

<ipxact:name>TNPPUT</ipxact:name>
                                                        </ipxact:logicalPort>
                                                        <ipxact:physicalPort>

<ipxact:name>csmea_cse_sb_TNPPUT</ipxact:name>
                                                        </ipxact:physicalPort>
                                                </ipxact:portMap>
                                                <ipxact:portMap>
                                                        <ipxact:logicalPort>

<ipxact:name>TPCPUT</ipxact:name>
                                                        </ipxact:logicalPort>
                                                        <ipxact:physicalPort>

<ipxact:name>csmea_cse_sb_TPCPUT</ipxact:name>
                                                        </ipxact:physicalPort>
                                                </ipxact:portMap>
                                                <ipxact:portMap>
                                                        <ipxact:logicalPort>
```

```
<ipxact:name>TNPCUP</ipxact:name>
                                              </ipxact:logicalPort>
                                              <ipxact:physicalPort>

<ipxact:name>csmea_cse_sb_TNPCUP</ipxact:name>
                                              </ipxact:physicalPort>
                                      </ipxact:portMap>
                                      <ipxact:portMap>
                                              <ipxact:logicalPort>

<ipxact:name>TPCCUP</ipxact:name>
                                              </ipxact:logicalPort>
                                              <ipxact:physicalPort>

<ipxact:name>csmea_cse_sb_TPCCUP</ipxact:name>
                                              </ipxact:physicalPort>
                                      </ipxact:portMap>
                                      <ipxact:portMap>
                                              <ipxact:logicalPort>

<ipxact:name>TEOM</ipxact:name>
                                              </ipxact:logicalPort>
                                              <ipxact:physicalPort>

<ipxact:name>csmea_cse_sb_TEOM</ipxact:name>
                                              </ipxact:physicalPort>
                                      </ipxact:portMap>
                                      <ipxact:portMap>
                                              <ipxact:logicalPort>

<ipxact:name>TPAYLOAD</ipxact:name>
                                              </ipxact:logicalPort>
                                              <ipxact:physicalPort>

<ipxact:name>csmea_cse_sb_TPAYLOAD</ipxact:name>
                                              </ipxact:physicalPort>
                                      </ipxact:portMap>
                                      <ipxact:portMap>
                                              <ipxact:logicalPort>

<ipxact:name>TPARITY</ipxact:name>
                                              </ipxact:logicalPort>
                                              <ipxact:physicalPort>

<ipxact:name>csmea_cse_sb_TPARITY_NOTSUPPORTED</ipxact:name>
                                              </ipxact:physicalPort>
                                      </ipxact:portMap>
                                      <ipxact:portMap>
                                              <ipxact:logicalPort>

<ipxact:name>SIDE_ISM_FABRIC</ipxact:name>
                                              </ipxact:logicalPort>
                                              <ipxact:physicalPort>

<ipxact:name>csmea_cse_sb_SIDE_ISM_FABRIC</ipxact:name>
                                              </ipxact:physicalPort>
                                      </ipxact:portMap>
                                      <ipxact:portMap>
                                              <ipxact:logicalPort>

<ipxact:name>SIDE_ISM_AGENT</ipxact:name>
                                              </ipxact:logicalPort>
                                              <ipxact:physicalPort>

<ipxact:name>csmea_cse_sb_SIDE_ISM_AGENT</ipxact:name>
                                              </ipxact:physicalPort>
```

```
                                                        </ipxact:portMap>
                                                </ipxact:portMaps>
                                        </ipxact:abstractionType>
                                </ipxact:abstractionTypes>
                                <ipxact:slave></ipxact:slave>
                                <ipxact:vendorExtensions>
                                        <intel:interface_attributes>
                                                <intel:is_sync>true</intel:is_sync>
                                                <intel:clk>side_clk</intel:clk>
                                                <intel:rst>my_resets</intel:rst>
                                        </intel:interface_attributes>
                                </ipxact:vendorExtensions>
                        </ipxact:busInterface>
                        <ipxact:busInterface>
                                <ipxact:name>CLOCK_REQ_ACK_Controller</ipxact:name>
                                <ipxact:description>Intended for IPs using Chassis 2.0 definition
of clk requests</ipxact:description>
                                <ipxact:busType vendor="Intel" library="Intel" name="CLOCK_REQ_ACK"
version="v2.0_r1.1"></ipxact:busType>
                                <ipxact:abstractionTypes>
                                        <ipxact:abstractionType>
                                                <ipxact:abstractionRef vendor="Intel"
library="Intel" name="CLOCK_REQ_ACK_rtl" version="v2.0_r1.1"></ipxact:abstractionRef>
                                                <ipxact:portMaps>
                                                        <ipxact:portMap>
                                                                <ipxact:logicalPort>

        <ipxact:name>CLKREQ</ipxact:name>
                                                                </ipxact:logicalPort>
                                                                <ipxact:physicalPort>

        <ipxact:name>csmea_cse_pgcb_clkreq</ipxact:name>
                                                                </ipxact:physicalPort>
                                                        </ipxact:portMap>
                                                        <ipxact:portMap>
                                                                <ipxact:logicalPort>

        <ipxact:name>CLKACK</ipxact:name>
                                                                </ipxact:logicalPort>
                                                                <ipxact:physicalPort>

        <ipxact:name>csmea_cse_pgcb_clkack</ipxact:name>
                                                                </ipxact:physicalPort>
                                                        </ipxact:portMap>
                                                </ipxact:portMaps>
                                        </ipxact:abstractionType>
                                </ipxact:abstractionTypes>
                                <ipxact:master></ipxact:master>
                                <ipxact:vendorExtensions>
                                        <intel:interface_attributes>
                                                <intel:is_sync>false</intel:is_sync>
                                        </intel:interface_attributes>
                                </ipxact:vendorExtensions>
                        </ipxact:busInterface>
                        <ipxact:busInterface>
                                <ipxact:name>SCAN_Consumer</ipxact:name>
                                <ipxact:description></ipxact:description>
                                <ipxact:busType vendor="Intel" library="Intel"
name="IOSF::DFX::SCAN" version="v2.5_r1.1"></ipxact:busType>
                                <ipxact:abstractionTypes>
                                        <ipxact:abstractionType>
                                                <ipxact:abstractionRef vendor="Intel"
library="Intel" name="IOSF::DFX::SCAN_rtl" version="v2.5_r1.1">
                                                        <ipxact:configurableElementValues>
                                                                <ipxact:configurableElementValue
referenceId="NUM_CLKGENCTRL">1</ipxact:configurableElementValue>
```

```
                                                    <ipxact:configurableElementValue
referenceId="NUM_CLKGENCTRLEN">1</ipxact:configurableElementValue>
                                                    <ipxact:configurableElementValue
referenceId="NUM_BYPRST_B">1</ipxact:configurableElementValue>
                                                    <ipxact:configurableElementValue
referenceId="SCAN_DATA_WIDTH">1</ipxact:configurableElementValue>
                                                    <ipxact:configurableElementValue
referenceId="NUM_RAM_BYPSEL">1</ipxact:configurableElementValue>
                                                    <ipxact:configurableElementValue
referenceId="NUM_BYPLATRST_B">1</ipxact:configurableElementValue>
                                                    <ipxact:configurableElementValue
referenceId="NUM_RSTBYPEN">1</ipxact:configurableElementValue>
                                            </ipxact:configurableElementValues>
                                    </ipxact:abstractionRef>
                                    <ipxact:portMaps>
                                            <ipxact:portMap>
                                                    <ipxact:logicalPort>

        <ipxact:name>FSCAN_SDI</ipxact:name>
                                                    </ipxact:logicalPort>
                                                    <ipxact:physicalPort>

        <ipxact:name>FSCAN_SDI_NOT_SUPPORTED</ipxact:name>
                                                    </ipxact:physicalPort>
                                            </ipxact:portMap>
                                            <ipxact:portMap>
                                                    <ipxact:logicalPort>

        <ipxact:name>ASCAN_SDO</ipxact:name>
                                                    </ipxact:logicalPort>
                                                    <ipxact:physicalPort>

        <ipxact:name>ASCAN_SDO_NOT_SUPPORTED</ipxact:name>
                                                    </ipxact:physicalPort>
                                            </ipxact:portMap>
                                            <ipxact:portMap>
                                                    <ipxact:logicalPort>

        <ipxact:name>FSCAN_MODE</ipxact:name>
                                                    </ipxact:logicalPort>
                                                    <ipxact:physicalPort>

        <ipxact:name>csmea_cse_fscan_mode</ipxact:name>
                                                    </ipxact:physicalPort>
                                            </ipxact:portMap>
                                            <ipxact:portMap>
                                                    <ipxact:logicalPort>

        <ipxact:name>FSCAN_MODE_ATSPEED</ipxact:name>
                                                    </ipxact:logicalPort>
                                                    <ipxact:physicalPort>

        <ipxact:name>FSCAN_MODE_ATSPEED_NOT_SUPPORTE</ipxact:name>
                                                    </ipxact:physicalPort>
                                            </ipxact:portMap>
                                            <ipxact:portMap>
                                                    <ipxact:logicalPort>

        <ipxact:name>FSCAN_RSTBYPEN</ipxact:name>
                                                    </ipxact:logicalPort>
                                                    <ipxact:physicalPort>

        <ipxact:name>csmea_cse_fscan_rstbypen</ipxact:name>
                                                    </ipxact:physicalPort>
                                            </ipxact:portMap>
                                            <ipxact:portMap>
                                                    <ipxact:logicalPort>
```

```
<ipxact:name>FSCAN_BYPRST_B</ipxact:name>
                                            </ipxact:logicalPort>
                                            <ipxact:physicalPort>

<ipxact:name>csmea_cse_fscan_byprst_b</ipxact:name>
                                            </ipxact:physicalPort>
                                    </ipxact:portMap>
                                    <ipxact:portMap>
                                            <ipxact:logicalPort>

<ipxact:name>FSCAN_BYPLATRST_B</ipxact:name>
                                            </ipxact:logicalPort>
                                            <ipxact:physicalPort>

<ipxact:name>csmea_cse_fscan_byplatrst_b</ipxact:name>
                                            </ipxact:physicalPort>
                                    </ipxact:portMap>
                                    <ipxact:portMap>
                                            <ipxact:logicalPort>

<ipxact:name>FSCAN_CLKUNGATE</ipxact:name>
                                            </ipxact:logicalPort>
                                            <ipxact:physicalPort>

<ipxact:name>csmea_cse_fscan_clkungate</ipxact:name>
                                            </ipxact:physicalPort>
                                    </ipxact:portMap>
                                    <ipxact:portMap>
                                            <ipxact:logicalPort>

<ipxact:name>FSCAN_CLKUNGATE_SYN</ipxact:name>
                                            </ipxact:logicalPort>
                                            <ipxact:physicalPort>

<ipxact:name>csmea_cse_fscan_clkungate_syn</ipxact:name>
                                            </ipxact:physicalPort>
                                    </ipxact:portMap>
                                    <ipxact:portMap>
                                            <ipxact:logicalPort>

<ipxact:name>FSCAN_LATCHOPEN</ipxact:name>
                                            </ipxact:logicalPort>
                                            <ipxact:physicalPort>

<ipxact:name>csmea_cse_fscan_latchopen</ipxact:name>
                                            </ipxact:physicalPort>
                                    </ipxact:portMap>
                                    <ipxact:portMap>
                                            <ipxact:logicalPort>

<ipxact:name>FSCAN_LATCHCLOSED_B</ipxact:name>
                                            </ipxact:logicalPort>
                                            <ipxact:physicalPort>

<ipxact:name>csmea_cse_fscan_latchclosed_b</ipxact:name>
                                            </ipxact:physicalPort>
                                    </ipxact:portMap>
                                    <ipxact:portMap>
                                            <ipxact:logicalPort>

<ipxact:name>FSCAN_CLKGENCTRL</ipxact:name>
                                            </ipxact:logicalPort>
                                            <ipxact:physicalPort>

<ipxact:name>csmea_cse_fscan_clkgenctrl</ipxact:name>
                                            </ipxact:physicalPort>
```

```
                                        </ipxact:portMap>
                                        <ipxact:portMap>
                                                <ipxact:logicalPort>

<ipxact:name>FSCAN_CLKGENCTRLEN</ipxact:name>
                                                </ipxact:logicalPort>
                                                <ipxact:physicalPort>

<ipxact:name>csmea_cse_fscan_clkgenctrlen</ipxact:name>
                                                </ipxact:physicalPort>
                                        </ipxact:portMap>
                                        <ipxact:portMap>
                                                <ipxact:logicalPort>

<ipxact:name>FSCAN_MODE_POSTSCC</ipxact:name>
                                                </ipxact:logicalPort>
                                                <ipxact:physicalPort>

<ipxact:name>csmea_cse_fscan_mode_postscc</ipxact:name>
                                                </ipxact:physicalPort>
                                        </ipxact:portMap>
                                        <ipxact:portMap>
                                                <ipxact:logicalPort>

<ipxact:name>FSCAN_INTEST</ipxact:name>
                                                </ipxact:logicalPort>
                                                <ipxact:physicalPort>

<ipxact:name>FSCAN_INTEST_NOT_SUPPORTED</ipxact:name>
                                                </ipxact:physicalPort>
                                        </ipxact:portMap>
                                        <ipxact:portMap>
                                                <ipxact:logicalPort>

<ipxact:name>FSCAN_EXTEST</ipxact:name>
                                                </ipxact:logicalPort>
                                                <ipxact:physicalPort>

<ipxact:name>FSCAN_EXTEST_NOT_SUPPORTED</ipxact:name>
                                                </ipxact:physicalPort>
                                        </ipxact:portMap>
                                        <ipxact:portMap>
                                                <ipxact:logicalPort>

<ipxact:name>FSCAN_RET_CTRL</ipxact:name>
                                                </ipxact:logicalPort>
                                                <ipxact:physicalPort>

<ipxact:name>fscan_ret_ctrl</ipxact:name>
                                                </ipxact:physicalPort>
                                        </ipxact:portMap>
                                        <ipxact:portMap>
                                                <ipxact:logicalPort>

<ipxact:name>FSCAN_ISOL_CTRL</ipxact:name>
                                                </ipxact:logicalPort>
                                                <ipxact:physicalPort>

<ipxact:name>csmea_cse_fscan_isol_ctrl</ipxact:name>
                                                </ipxact:physicalPort>
                                        </ipxact:portMap>
                                        <ipxact:portMap>
                                                <ipxact:logicalPort>

<ipxact:name>FSCAN_ISOL_LAT_CTRL</ipxact:name>
                                                </ipxact:logicalPort>
                                                <ipxact:physicalPort>
```

```
<ipxact:name>csmea_cse_fscan_isol_lat_ctrl</ipxact:name>
                                             </ipxact:physicalPort>
                                      </ipxact:portMap>
                                      <ipxact:portMap>
                                             <ipxact:logicalPort>

<ipxact:name>FSCAN_SHIFTEN</ipxact:name>
                                             </ipxact:logicalPort>
                                             <ipxact:physicalPort>

<ipxact:name>csmea_cse_fscan_shiften</ipxact:name>
                                             </ipxact:physicalPort>
                                      </ipxact:portMap>
                                      <ipxact:portMap>
                                             <ipxact:logicalPort>

<ipxact:name>FSCAN_SLOS_EN</ipxact:name>
                                             </ipxact:logicalPort>
                                             <ipxact:physicalPort>

<ipxact:name>FSCAN_SLOS_EN_NOT_SUPPORTED</ipxact:name>
                                             </ipxact:physicalPort>
                                      </ipxact:portMap>
                                      <ipxact:portMap>
                                             <ipxact:logicalPort>

<ipxact:name>FSCAN_TPI_CONTROL_EN</ipxact:name>
                                             </ipxact:logicalPort>
                                             <ipxact:physicalPort>

<ipxact:name>FSCAN_TPI_CONTROL_EN_NOT_SUPPORTED</ipxact:name>
                                             </ipxact:physicalPort>
                                      </ipxact:portMap>
                                      <ipxact:portMap>
                                             <ipxact:logicalPort>

<ipxact:name>FSCAN_TPI_OBSERVE_EN</ipxact:name>
                                             </ipxact:logicalPort>
                                             <ipxact:physicalPort>

<ipxact:name>FSCAN_TPI_OBSERVE_EN_NOT_SUPPORTED</ipxact:name>
                                             </ipxact:physicalPort>
                                      </ipxact:portMap>
                                      <ipxact:portMap>
                                             <ipxact:logicalPort>

<ipxact:name>FSCAN_RAM_BYPSEL</ipxact:name>
                                             </ipxact:logicalPort>
                                             <ipxact:physicalPort>

<ipxact:name>csmea_cse_fscan_ram_bypsel</ipxact:name>
                                             </ipxact:physicalPort>
                                      </ipxact:portMap>
                                      <ipxact:portMap>
                                             <ipxact:logicalPort>

<ipxact:name>FSCAN_RAM_INIT_EN</ipxact:name>
                                             </ipxact:logicalPort>
                                             <ipxact:physicalPort>

<ipxact:name>FSCAN_RAM_INIT_EN_NOT_SUPPORTED</ipxact:name>
                                             </ipxact:physicalPort>
                                      </ipxact:portMap>
                                      <ipxact:portMap>
                                             <ipxact:logicalPort>
```

```
        <ipxact:name>FSCAN_RAM_INIT_VAL</ipxact:name>
                                                        </ipxact:logicalPort>
                                                        <ipxact:physicalPort>

        <ipxact:name>FSCAN_RAM_INIT_VAL_NOT_SUPPORTED</ipxact:name>
                                                        </ipxact:physicalPort>
                                                </ipxact:portMap>
                                                <ipxact:portMap>
                                                        <ipxact:logicalPort>

        <ipxact:name>FSCAN_RAM_RDDIS_B</ipxact:name>
                                                        </ipxact:logicalPort>
                                                        <ipxact:physicalPort>

        <ipxact:name>csmea_cse_fscan_ram_rddis_b</ipxact:name>
                                                        </ipxact:physicalPort>
                                                </ipxact:portMap>
                                                <ipxact:portMap>
                                                        <ipxact:logicalPort>

        <ipxact:name>FSCAN_RAM_WRDIS_B</ipxact:name>
                                                        </ipxact:logicalPort>
                                                        <ipxact:physicalPort>

        <ipxact:name>csmea_cse_fscan_ram_wrdis_b</ipxact:name>
                                                        </ipxact:physicalPort>
                                                </ipxact:portMap>
                                        </ipxact:portMaps>
                                </ipxact:abstractionType>
                        </ipxact:abstractionTypes>
                        <ipxact:slave></ipxact:slave>
                        <ipxact:vendorExtensions>
                                <intel:interface_attributes>
                                        <intel:is_sync>false</intel:is_sync>
                                        <intel:rst>my_resets</intel:rst>
                                </intel:interface_attributes>
                        </ipxact:vendorExtensions>
                </ipxact:busInterface>
        </ipxact:busInterfaces>
        <ipxact:model>
                <ipxact:views>
                        <ipxact:view>
                                <ipxact:name>RTL</ipxact:name>

        <ipxact:componentInstantiationRef>RTL</ipxact:componentInstantiationRef>
                        </ipxact:view>
                </ipxact:views>
                <ipxact:instantiations>
                        <ipxact:componentInstantiation>
                                <ipxact:name>RTL</ipxact:name>
                                <ipxact:description>IP RTL Wrapper
Module.</ipxact:description>
                                <ipxact:language>systemverilog</ipxact:language>
                                <ipxact:moduleName>NickTest1</ipxact:moduleName>
                        </ipxact:componentInstantiation>
                </ipxact:instantiations>
                <ipxact:ports>
                        <ipxact:port>
                                <ipxact:name>NS1</ipxact:name>
                                <ipxact:wire>
                                        <ipxact:direction>out</ipxact:direction>
                                </ipxact:wire>
                                <ipxact:vendorExtensions>
                                        <intel:port_attributes>

        <intel:clk_domain>side_clk</intel:clk_domain>
        <intel:voltage_domain>my_powerdomains</intel:voltage_domain>
                                        </intel:port_attributes>
                                </ipxact:vendorExtensions>
                        </ipxact:port>
                        <ipxact:port>
                                <ipxact:name>NS2</ipxact:name>
                                <ipxact:wire>
                                        <ipxact:direction>in</ipxact:direction>
                                </ipxact:wire>
                                <ipxact:vendorExtensions>
                                        <intel:port_attributes>

        <intel:clk_domain>side_clk</intel:clk_domain>

        <intel:voltage_domain>my_powerdomains</intel:voltage_domain>
                                        </intel:port_attributes>
                                </ipxact:vendorExtensions>
                        </ipxact:port>
                        <ipxact:port>
                                <ipxact:name>foo_rst</ipxact:name>
                                <ipxact:wire>
                                        <ipxact:direction>in</ipxact:direction>
                                </ipxact:wire>
                                <ipxact:vendorExtensions>
                                        <intel:port_attributes>
                                                <intel:clk_domain>clk1</intel:clk_domain>

        <intel:voltage_domain>warm</intel:voltage_domain>
                                        </intel:port_attributes>
                                </ipxact:vendorExtensions>
                        </ipxact:port>
                        <ipxact:port>
                                <ipxact:name>bar_rst</ipxact:name>
                                <ipxact:wire>
                                        <ipxact:direction>out</ipxact:direction>
                                </ipxact:wire>
                                <ipxact:vendorExtensions>
                                        <intel:port_attributes>
                                                <intel:clk_domain>clk1</intel:clk_domain>

        <intel:voltage_domain>warm</intel:voltage_domain>
                                        </intel:port_attributes>
                                </ipxact:vendorExtensions>
                        </ipxact:port>
                        <ipxact:port>
                                <ipxact:name>cfmia_fdfx_rst_b</ipxact:name>
                                <ipxact:description>Ports for Manually exported
pins</ipxact:description>
                                <ipxact:wire>
                                        <ipxact:direction>in</ipxact:direction>
                                </ipxact:wire>
                                <ipxact:vendorExtensions>
                                        <intel:port_attributes>
                                                <intel:clk_domain>clk1</intel:clk_domain>

        <intel:voltage_domain>warm</intel:voltage_domain>
                                        </intel:port_attributes>
                                </ipxact:vendorExtensions>
                        </ipxact:port>
                        <ipxact:port>

        <ipxact:name>cse_cfmia_adhoc_intf_cfmia_cse_mia_cfi_cse_rstisol_ack</ipxact:name>
                                <ipxact:description>parameter from RTL file
csmea_cfmia.sv</ipxact:description>
                                <ipxact:wire>
                                        <ipxact:direction>out</ipxact:direction>
```

```
                              </ipxact:wire>
                              <ipxact:vendorExtensions>
                                     <intel:port_attributes>
                                            <intel:clk_domain>clk1</intel:clk_domain>

       <intel:voltage_domain>warm</intel:voltage_domain>
                                     </intel:port_attributes>
                              </ipxact:vendorExtensions>
                       </ipxact:port>
                       <ipxact:port>

       <ipxact:name>cse_cfmia_adhoc_intf_cse_cfmia_cfi_mia_cse_rstisol_req</ipxact:name>
                              <ipxact:description>parameter from RTL file
csmea_cfmia.sv</ipxact:description>
                              <ipxact:wire>
                                     <ipxact:direction>in</ipxact:direction>
                              </ipxact:wire>
                              <ipxact:vendorExtensions>
                                     <intel:port_attributes>
                                            <intel:clk_domain>clk1</intel:clk_domain>

       <intel:voltage_domain>warm</intel:voltage_domain>
                                     </intel:port_attributes>
                              </ipxact:vendorExtensions>
                       </ipxact:port>
                       <ipxact:port>

       <ipxact:name>cse_cfmia_adhoc_intf_cse_cfmia_cse_rst_b</ipxact:name>
                              <ipxact:description>parameter from RTL file
csmea_cfmia.sv</ipxact:description>
                              <ipxact:wire>
                                     <ipxact:direction>in</ipxact:direction>
                              </ipxact:wire>
                              <ipxact:vendorExtensions>
                                     <intel:port_attributes>
                                            <intel:clk_domain>clk1</intel:clk_domain>

       <intel:voltage_domain>warm</intel:voltage_domain>
                                     </intel:port_attributes>
                              </ipxact:vendorExtensions>
                       </ipxact:port>
                       <ipxact:port>

       <ipxact:name>cse_cfmia_adhoc_intf_cse_cfmia_pg_sidefunc_rst_b</ipxact:name>
                              <ipxact:description>parameter from RTL file
csmea_cfmia.sv</ipxact:description>
                              <ipxact:wire>
                                     <ipxact:direction>in</ipxact:direction>
                              </ipxact:wire>
                              <ipxact:vendorExtensions>
                                     <intel:port_attributes>
                                            <intel:clk_domain>clk1</intel:clk_domain>

       <intel:voltage_domain>warm</intel:voltage_domain>
                                     </intel:port_attributes>
                              </ipxact:vendorExtensions>
                       </ipxact:port>
                       <ipxact:port>

       <ipxact:name>cse_cfmia_adhoc_intf_cse_cfmia_pgcb_force_rst_b</ipxact:name>
                              <ipxact:description>parameter from RTL file
csmea_cfmia.sv</ipxact:description>
                              <ipxact:wire>
                                     <ipxact:direction>in</ipxact:direction>
                              </ipxact:wire>
                              <ipxact:vendorExtensions>
                                     <intel:port_attributes>
```

```
                                                            <intel:clk_domain>clk1</intel:clk_domain>

        <intel:voltage_domain>warm</intel:voltage_domain>
                                                    </intel:port_attributes>
                                            </ipxact:vendorExtensions>
                            </ipxact:port>
                            <ipxact:port>

        <ipxact:name>cse_cfmia_adhoc_intf_cse_cfmia_pgcb_rst_b</ipxact:name>
                                            <ipxact:description>parameter from RTL file
csmea_cfmia.sv</ipxact:description>
                                            <ipxact:wire>
                                                    <ipxact:direction>in</ipxact:direction>
                                            </ipxact:wire>
                                            <ipxact:vendorExtensions>
                                                    <intel:port_attributes>
                                                            <intel:clk_domain>clk1</intel:clk_domain>

        <intel:voltage_domain>warm</intel:voltage_domain>
                                                    </intel:port_attributes>
                                            </ipxact:vendorExtensions>
                            </ipxact:port>
                            <ipxact:port>

        <ipxact:name>cse_cfmia_adhoc_intf_cse_cfmia_prim_rst_b</ipxact:name>
                                            <ipxact:description>parameter from RTL file
csmea_cfmia.sv</ipxact:description>
                                            <ipxact:wire>
                                                    <ipxact:direction>in</ipxact:direction>
                                            </ipxact:wire>
                                            <ipxact:vendorExtensions>
                                                    <intel:port_attributes>
                                                            <intel:clk_domain>clk1</intel:clk_domain>

        <intel:voltage_domain>warm</intel:voltage_domain>
                                                    </intel:port_attributes>
                                            </ipxact:vendorExtensions>
                            </ipxact:port>
                            <ipxact:port>

        <ipxact:name>cse_cfmia_adhoc_intf_cse_cfmia_pwrok_rst_b</ipxact:name>
                                            <ipxact:description>parameter from RTL file
csmea_cfmia.sv</ipxact:description>
                                            <ipxact:wire>
                                                    <ipxact:direction>in</ipxact:direction>
                                            </ipxact:wire>
                                            <ipxact:vendorExtensions>
                                                    <intel:port_attributes>
                                                            <intel:clk_domain>clk1</intel:clk_domain>

        <intel:voltage_domain>warm</intel:voltage_domain>
                                                    </intel:port_attributes>
                                            </ipxact:vendorExtensions>
                            </ipxact:port>
                            <ipxact:port>

        <ipxact:name>cse_cfmia_adhoc_intf_cse_cfmia_rstsync_clken</ipxact:name>
                                            <ipxact:description>parameter from RTL file
csmea_cfmia.sv</ipxact:description>
                                            <ipxact:wire>
                                                    <ipxact:direction>in</ipxact:direction>
                                            </ipxact:wire>
                                            <ipxact:vendorExtensions>
                                                    <intel:port_attributes>
                                                            <intel:clk_domain>clk1</intel:clk_domain>

        <intel:voltage_domain>warm</intel:voltage_domain>
```

```
                                        </intel:port_attributes>
                                </ipxact:vendorExtensions>
                        </ipxact:port>
                        <ipxact:port>
                                <ipxact:name>cse_fdfx_pwrgood_rst_b</ipxact:name>
                                <ipxact:description>Ports for Manually exported
pins</ipxact:description>
                                <ipxact:wire>
                                        <ipxact:direction>in</ipxact:direction>
                                </ipxact:wire>
                                <ipxact:vendorExtensions>
                                        <intel:port_attributes>
                                                <intel:clk_domain>clk1</intel:clk_domain>

        <intel:voltage_domain>warm</intel:voltage_domain>
                                        </intel:port_attributes>
                                </ipxact:vendorExtensions>
                        </ipxact:port>
                        <ipxact:port>

        <ipxact:name>csmea_cfmia_cfmia_fary_bisr_reset_rf</ipxact:name>
                                <ipxact:description>parameter from RTL file
csmea_cfmia.sv</ipxact:description>
                                <ipxact:wire>
                                        <ipxact:direction>in</ipxact:direction>
                                </ipxact:wire>
                                <ipxact:vendorExtensions>
                                        <intel:port_attributes>
                                                <intel:clk_domain>clk1</intel:clk_domain>

        <intel:voltage_domain>warm</intel:voltage_domain>
                                        </intel:port_attributes>
                                </ipxact:vendorExtensions>
                        </ipxact:port>
                        <ipxact:port>
                                <ipxact:name>csmea_cfmia_fscan_byplatrst_b</ipxact:name>
                                <ipxact:description>Ports for Manually exported
pins</ipxact:description>
                                <ipxact:wire>
                                        <ipxact:direction>in</ipxact:direction>
                                </ipxact:wire>
                                <ipxact:vendorExtensions>
                                        <intel:port_attributes>
                                                <intel:clk_domain>clk1</intel:clk_domain>

        <intel:voltage_domain>warm</intel:voltage_domain>
                                        </intel:port_attributes>
                                </ipxact:vendorExtensions>
                        </ipxact:port>
                        <ipxact:port>
                                <ipxact:name>csmea_cfmia_fscan_byprst_b</ipxact:name>
                                <ipxact:description>Ports for Manually exported
pins</ipxact:description>
                                <ipxact:wire>
                                        <ipxact:direction>in</ipxact:direction>
                                </ipxact:wire>
                                <ipxact:vendorExtensions>
                                        <intel:port_attributes>
                                                <intel:clk_domain>clk1</intel:clk_domain>

        <intel:voltage_domain>warm</intel:voltage_domain>
                                        </intel:port_attributes>
                                </ipxact:vendorExtensions>
                        </ipxact:port>
                        <ipxact:port>
                                <ipxact:name>csmea_cfmia_fscan_rstbypen</ipxact:name>
```

```
                              <ipxact:description>Ports for Manually exported
pins</ipxact:description>
                              <ipxact:wire>
                                     <ipxact:direction>in</ipxact:direction>
                              </ipxact:wire>
                              <ipxact:vendorExtensions>
                                     <intel:port_attributes>
                                            <intel:clk_domain>clk1</intel:clk_domain>

       <intel:voltage_domain>warm</intel:voltage_domain>
                                     </intel:port_attributes>
                              </ipxact:vendorExtensions>
                       </ipxact:port>
                       <ipxact:port>

       <ipxact:name>csmea_cfmia_sramc_fary_bisr_reset</ipxact:name>
                              <ipxact:description>parameter from RTL file
csmea_cfmia.sv</ipxact:description>
                              <ipxact:wire>
                                     <ipxact:direction>in</ipxact:direction>
                              </ipxact:wire>
                              <ipxact:vendorExtensions>
                                     <intel:port_attributes>
                                            <intel:clk_domain>clk1</intel:clk_domain>

       <intel:voltage_domain>warm</intel:voltage_domain>
                                     </intel:port_attributes>
                              </ipxact:vendorExtensions>
                       </ipxact:port>
                       <ipxact:port>
                              <ipxact:name>moa_cse_cfmia_cfn_mia_rst_n_hclk</ipxact:name>
                              <ipxact:description>Ports for Manually exported
pins</ipxact:description>
                              <ipxact:wire>
                                     <ipxact:direction>out</ipxact:direction>
                              </ipxact:wire>
                              <ipxact:vendorExtensions>
                                     <intel:port_attributes>
                                            <intel:clk_domain>clk1</intel:clk_domain>

       <intel:voltage_domain>warm</intel:voltage_domain>
                                     </intel:port_attributes>
                              </ipxact:vendorExtensions>
                       </ipxact:port>
                       <ipxact:port>
                              <ipxact:name>csmea_cse_sb_MNPPUT</ipxact:name>
                              <ipxact:description>Non-Posted Put from endpoint to router
</ipxact:description>
                              <ipxact:wire>
                                     <ipxact:direction>out</ipxact:direction>
                              </ipxact:wire>
                              <ipxact:vendorExtensions>
                                     <intel:port_attributes>

       <intel:clk_domain>side_clk</intel:clk_domain>

       <intel:voltage_domain>my_powerdomains</intel:voltage_domain>
                                     </intel:port_attributes>
                              </ipxact:vendorExtensions>
                       </ipxact:port>
                       <ipxact:port>
                              <ipxact:name>csmea_cse_sb_MPCPUT</ipxact:name>
                              <ipxact:description>Posted or Completion Put from endpoint
to router</ipxact:description>
                              <ipxact:wire>
                                     <ipxact:direction>out</ipxact:direction>
                              </ipxact:wire>
                              <ipxact:vendorExtensions>
                                      <intel:port_attributes>

        <intel:clk_domain>side_clk</intel:clk_domain>

        <intel:voltage_domain>my_powerdomains</intel:voltage_domain>
                                      </intel:port_attributes>
                              </ipxact:vendorExtensions>
                      </ipxact:port>
                      <ipxact:port>
                              <ipxact:name>csmea_cse_sb_MNPCUP</ipxact:name>
                              <ipxact:description>Non-Posted Credit Update from router to
endpoint</ipxact:description>
                              <ipxact:wire>
                                      <ipxact:direction>in</ipxact:direction>
                              </ipxact:wire>
                              <ipxact:vendorExtensions>
                                      <intel:port_attributes>

        <intel:clk_domain>side_clk</intel:clk_domain>

        <intel:voltage_domain>my_powerdomains</intel:voltage_domain>
                                      </intel:port_attributes>
                              </ipxact:vendorExtensions>
                      </ipxact:port>
                      <ipxact:port>
                              <ipxact:name>csmea_cse_sb_MPCCUP</ipxact:name>
                              <ipxact:description>Posted or Completion Credit Update from
router to endpoint</ipxact:description>
                              <ipxact:wire>
                                      <ipxact:direction>in</ipxact:direction>
                              </ipxact:wire>
                              <ipxact:vendorExtensions>
                                      <intel:port_attributes>

        <intel:clk_domain>side_clk</intel:clk_domain>

        <intel:voltage_domain>my_powerdomains</intel:voltage_domain>
                                      </intel:port_attributes>
                              </ipxact:vendorExtensions>
                      </ipxact:port>
                      <ipxact:port>
                              <ipxact:name>csmea_cse_sb_MEOM</ipxact:name>
                              <ipxact:description>End of Message from endpoint to
router</ipxact:description>
                              <ipxact:wire>
                                      <ipxact:direction>out</ipxact:direction>
                              </ipxact:wire>
                              <ipxact:vendorExtensions>
                                      <intel:port_attributes>

        <intel:clk_domain>side_clk</intel:clk_domain>

        <intel:voltage_domain>my_powerdomains</intel:voltage_domain>
                                      </intel:port_attributes>
                              </ipxact:vendorExtensions>
                      </ipxact:port>
                      <ipxact:port>
                              <ipxact:name>csmea_cse_sb_MPAYLOAD</ipxact:name>
                              <ipxact:description>Message Payload from endpoint to
router</ipxact:description>
                              <ipxact:wire>
                                      <ipxact:direction>out</ipxact:direction>
                                      <ipxact:vectors>
                                              <ipxact:vector>
                                                      <ipxact:left>7</ipxact:left>
                                                      <ipxact:right>0</ipxact:right>
```

```
                                                </ipxact:vector>
                                        </ipxact:vectors>
                                </ipxact:wire>
                                <ipxact:vendorExtensions>
                                        <intel:port_attributes>

        <intel:clk_domain>side_clk</intel:clk_domain>

        <intel:voltage_domain>my_powerdomains</intel:voltage_domain>
                                        </intel:port_attributes>
                                </ipxact:vendorExtensions>
                        </ipxact:port>
                        <ipxact:port>

        <ipxact:name>csmea_cse_sb_MPARITY_NOTSUPPORTED</ipxact:name>
                                <ipxact:description>Message parity from endpoint to
router</ipxact:description>
                                <ipxact:wire>
                                        <ipxact:direction>out</ipxact:direction>
                                </ipxact:wire>
                                <ipxact:vendorExtensions>
                                        <intel:port_attributes>

        <intel:clk_domain>side_clk</intel:clk_domain>

        <intel:voltage_domain>my_powerdomains</intel:voltage_domain>
                                        </intel:port_attributes>
                                </ipxact:vendorExtensions>
                        </ipxact:port>
                        <ipxact:port>
                                <ipxact:name>csmea_cse_sb_TNPPUT</ipxact:name>
                                <ipxact:description>Non-Posted Put from router to
endpoint</ipxact:description>
                                <ipxact:wire>
                                        <ipxact:direction>in</ipxact:direction>
                                </ipxact:wire>
                                <ipxact:vendorExtensions>
                                        <intel:port_attributes>

        <intel:clk_domain>side_clk</intel:clk_domain>

        <intel:voltage_domain>my_powerdomains</intel:voltage_domain>
                                        </intel:port_attributes>
                                </ipxact:vendorExtensions>
                        </ipxact:port>
                        <ipxact:port>
                                <ipxact:name>csmea_cse_sb_TPCPUT</ipxact:name>
                                <ipxact:description>Posted or Completion Put from router to
endpoint</ipxact:description>
                                <ipxact:wire>
                                        <ipxact:direction>in</ipxact:direction>
                                </ipxact:wire>
                                <ipxact:vendorExtensions>
                                        <intel:port_attributes>

        <intel:clk_domain>side_clk</intel:clk_domain>

        <intel:voltage_domain>my_powerdomains</intel:voltage_domain>
                                        </intel:port_attributes>
                                </ipxact:vendorExtensions>
                        </ipxact:port>
                        <ipxact:port>
                                <ipxact:name>csmea_cse_sb_TNPCUP</ipxact:name>
                                <ipxact:description>Non-Posted Credit Update from endpoint
to router</ipxact:description>
                                <ipxact:wire>
                                        <ipxact:direction>out</ipxact:direction>
```

```
                              </ipxact:wire>
                              <ipxact:vendorExtensions>
                                     <intel:port_attributes>

      <intel:clk_domain>side_clk</intel:clk_domain>

      <intel:voltage_domain>my_powerdomains</intel:voltage_domain>
                                     </intel:port_attributes>
                              </ipxact:vendorExtensions>
                       </ipxact:port>
                       <ipxact:port>
                              <ipxact:name>csmea_cse_sb_TPCCUP</ipxact:name>
                              <ipxact:description>Posted or Completion Credit Update from
endpoint to router</ipxact:description>
                              <ipxact:wire>
                                     <ipxact:direction>out</ipxact:direction>
                              </ipxact:wire>
                              <ipxact:vendorExtensions>
                                     <intel:port_attributes>

      <intel:clk_domain>side_clk</intel:clk_domain>

      <intel:voltage_domain>my_powerdomains</intel:voltage_domain>
                                     </intel:port_attributes>
                              </ipxact:vendorExtensions>
                       </ipxact:port>
                       <ipxact:port>
                              <ipxact:name>csmea_cse_sb_TEOM</ipxact:name>
                              <ipxact:description>End of Message from router to
endpoint</ipxact:description>
                              <ipxact:wire>
                                     <ipxact:direction>in</ipxact:direction>
                              </ipxact:wire>
                              <ipxact:vendorExtensions>
                                     <intel:port_attributes>

      <intel:clk_domain>side_clk</intel:clk_domain>

      <intel:voltage_domain>my_powerdomains</intel:voltage_domain>
                                     </intel:port_attributes>
                              </ipxact:vendorExtensions>
                       </ipxact:port>
                       <ipxact:port>
                              <ipxact:name>csmea_cse_sb_TPAYLOAD</ipxact:name>
                              <ipxact:description>Message Payload from router to
endpoint</ipxact:description>
                              <ipxact:wire>
                                     <ipxact:direction>in</ipxact:direction>
                                     <ipxact:vectors>
                                            <ipxact:vector>
                                                   <ipxact:left>7</ipxact:left>
                                                   <ipxact:right>0</ipxact:right>
                                            </ipxact:vector>
                                     </ipxact:vectors>
                              </ipxact:wire>
                              <ipxact:vendorExtensions>
                                     <intel:port_attributes>

      <intel:clk_domain>side_clk</intel:clk_domain>

      <intel:voltage_domain>my_powerdomains</intel:voltage_domain>
                                     </intel:port_attributes>
                              </ipxact:vendorExtensions>
                       </ipxact:port>
                       <ipxact:port>

      <ipxact:name>csmea_cse_sb_TPARITY_NOTSUPPORTED</ipxact:name>
```

```
                              <ipxact:description>Message parity from router to
endpoint</ipxact:description>
                              <ipxact:wire>
                                     <ipxact:direction>in</ipxact:direction>
                              </ipxact:wire>
                              <ipxact:vendorExtensions>
                                     <intel:port_attributes>

       <intel:clk_domain>side_clk</intel:clk_domain>

       <intel:voltage_domain>my_powerdomains</intel:voltage_domain>
                                     </intel:port_attributes>
                              </ipxact:vendorExtensions>
                       </ipxact:port>
                       <ipxact:port>
                              <ipxact:name>csmea_cse_sb_SIDE_ISM_FABRIC</ipxact:name>
                              <ipxact:description>Sideband Fabric Idle State Machine:
Fabric-to-IOSF agent Idle State Machine</ipxact:description>
                              <ipxact:wire>
                                     <ipxact:direction>in</ipxact:direction>
                                     <ipxact:vectors>
                                            <ipxact:vector>
                                                   <ipxact:left>2</ipxact:left>
                                                   <ipxact:right>0</ipxact:right>
                                            </ipxact:vector>
                                     </ipxact:vectors>
                              </ipxact:wire>
                              <ipxact:vendorExtensions>
                                     <intel:port_attributes>

       <intel:clk_domain>side_clk</intel:clk_domain>

       <intel:voltage_domain>my_powerdomains</intel:voltage_domain>
                                     </intel:port_attributes>
                              </ipxact:vendorExtensions>
                       </ipxact:port>
                       <ipxact:port>
                              <ipxact:name>csmea_cse_sb_SIDE_ISM_AGENT</ipxact:name>
                              <ipxact:description>Sideband EP Idle State
Machine</ipxact:description>
                              <ipxact:wire>
                                     <ipxact:direction>out</ipxact:direction>
                                     <ipxact:vectors>
                                            <ipxact:vector>
                                                   <ipxact:left>2</ipxact:left>
                                                   <ipxact:right>0</ipxact:right>
                                            </ipxact:vector>
                                     </ipxact:vectors>
                              </ipxact:wire>
                              <ipxact:vendorExtensions>
                                     <intel:port_attributes>

       <intel:clk_domain>side_clk</intel:clk_domain>

       <intel:voltage_domain>my_powerdomains</intel:voltage_domain>
                                     </intel:port_attributes>
                              </ipxact:vendorExtensions>
                       </ipxact:port>
                       <ipxact:port>
                              <ipxact:name>csmea_cse_pgcb_clkreq</ipxact:name>
                              <ipxact:description>This signal is asserted by the IP to
indicate that it needs a clock</ipxact:description>
                              <ipxact:wire>
                                     <ipxact:direction>in</ipxact:direction>
                              </ipxact:wire>
                       </ipxact:port>
                       <ipxact:port>
```

```
                              <ipxact:name>csmea_cse_pgcb_clkack</ipxact:name>
                              <ipxact:description>This signal is asserted by the SOC to
indicate to the IP that clock is available</ipxact:description>
                              <ipxact:wire>
                                     <ipxact:direction>out</ipxact:direction>
                              </ipxact:wire>
                       </ipxact:port>
                       <ipxact:port>
                              <ipxact:name>FSCAN_SDI_NOT_SUPPORTED</ipxact:name>
                              <ipxact:description>Fabric Scan Data In: This signal bus is
the scan data inputs for all of the serially-stitched scan flops/latches within this IP-
agent.</ipxact:description>
                              <ipxact:wire>
                                     <ipxact:direction>in</ipxact:direction>
                              </ipxact:wire>
                       </ipxact:port>
                       <ipxact:port>
                              <ipxact:name>ASCAN_SDO_NOT_SUPPORTED</ipxact:name>
                              <ipxact:description>Agent Scan Data Out: This signal bus is
the scan data outputs for all of the serially-stitched scan flops/latches within this
agent. </ipxact:description>
                              <ipxact:wire>
                                     <ipxact:direction>out</ipxact:direction>
                              </ipxact:wire>
                       </ipxact:port>
                       <ipxact:port>
                              <ipxact:name>csmea_cse_fscan_mode</ipxact:name>
                              <ipxact:description>Fabric Scan Mode: This signal enables
modes within this agent for scan operations. This signal is bused to support a hard IP-
block’s physical layer that requires scan control per lane.

Use of a bit vector for this signal is optional. The value ScanCtlWidth is implementation
dependent and may vary per-agent or per-lane.

Soft-IP use model:This signal may or may not be used depending on the attributes within
the IP-block that need to be made scan friendly. However, it is still required on the
interface.

Hard-IP use model: Its primary use is to enable the UCC /URC controller for this hard-IP
partition. Other scan enabling features should be controlled by the asynchronous control
signal group. If a scan attribute is unique to this partition and a corresponding control
signal is not available than a local TAP test/debug register will enable its actions.
</ipxact:description>
                              <ipxact:wire>
                                     <ipxact:direction>in</ipxact:direction>
                              </ipxact:wire>
                       </ipxact:port>
                       <ipxact:port>
                              <ipxact:name>FSCAN_MODE_ATSPEED_NOT_SUPPORTE</ipxact:name>
                              <ipxact:description>Fabric Scan At-speed Mode: This signal
enables the at -speed mode for this agent. This signal is bused to support hard-IP block
modular physical layer that requires scan control per lane.

Use of a bit vector for this signal name is optional. The value of ScanCtlWidth is
implementation-dependent and may vary per-agent or per-lane.

Note:This signal is marked as optional, but it should always be implemented unless the IP
has a very specific waiver for not requiring at-speed scan content.</ipxact:description>
                              <ipxact:wire>
                                     <ipxact:direction>in</ipxact:direction>
                              </ipxact:wire>
                       </ipxact:port>
                       <ipxact:port>
                              <ipxact:name>csmea_cse_fscan_rstbypen</ipxact:name>
                              <ipxact:description>Fabric Scan Reset Bypass Enable: This
signal will enable the ability for the bypass reset signals to be active. The reset
```

override signal group must be implemented for IP-blocks with embedded or derived internal
reset signals.

0: Reset bypass and Latch reset bypass are ignored.
1: Reset bypass and Latch reset bypass are active.

Use of a bit vector for this signal name is optional. The value of NumOfRstsScan is
implementation-dependent and may vary per-agent or per-lane. This signal is bused to
support hard-IP block modular physical layer that requires scan control per lane.

Note:It is expected that a soft-IP agent will only use a single control wire for enabling
the reset bypasses. A hard-IP agent that implements scan on a per lane basis will require
a vector set of enable signals.</ipxact:description>
                              <ipxact:wire>
                                      <ipxact:direction>in</ipxact:direction>
                              </ipxact:wire>
                      </ipxact:port>
                      <ipxact:port>
                              <ipxact:name>csmea_cse_fscan_byprst_b</ipxact:name>
                              <ipxact:description>Fabric Scan Bypass Reset Bar: This
signal is a reset input for scan operations that bypasses the internal agent reset logic
and applies a reset directly to the agent.The reset override signal group must be
implemented for IP-blocks with embedded or derived internal reset signals.

Note:Use of a bit vector for this signal name is optional. The value of NumOfRstsScan is
implementation-dependent and may vary per-agent or per-lane.

Note:This signal is enabled with fscan_rstbypen.</ipxact:description>
                              <ipxact:wire>
                                      <ipxact:direction>in</ipxact:direction>
                              </ipxact:wire>
                      </ipxact:port>
                      <ipxact:port>
                              <ipxact:name>csmea_cse_fscan_byplatrst_b</ipxact:name>
                              <ipxact:description>Fabric Scan Bypass Latch Reset Bar:
This signal is a reset input for scan operations that bypasses the internal agent reset
logic and applies a reset directly to the latches within the agent. This signal is bused
to support the hard-IP block modular physical layer that requires scan control per lane.

Note:Use of a bit vector for this signal name is optional. The value of NumOfLatRstsScan
is implementation-dependent and may vary per-agent or per-lane.

Note:This signal is enabled with fscan_rstbypen.</ipxact:description>
                              <ipxact:wire>
                                      <ipxact:direction>in</ipxact:direction>
                              </ipxact:wire>
                      </ipxact:port>
                      <ipxact:port>
                              <ipxact:name>csmea_cse_fscan_clkungate</ipxact:name>
                              <ipxact:description>Fabric Scan Clock Ungate: This signal
controls the clock gating logic during scan operations. This signal is bused to support
hard-IP block modular physical layer that requires scan control per
lane.</ipxact:description>
                              <ipxact:wire>
                                      <ipxact:direction>in</ipxact:direction>
                              </ipxact:wire>
                      </ipxact:port>
                      <ipxact:port>
                              <ipxact:name>csmea_cse_fscan_clkungate_syn</ipxact:name>
                              <ipxact:description>Fabric Scan Clock Ungate for Synthesis
Inserted Clock Gates: This signal controls the clock gating logic inserted during
synthesis. This signal cannot be used interchangeably with the fscan_clkungate signal
that is used exclusively to control clock gating logic that exists in the pre-synthesis
design. This signal controls the clock gating logic that is added after synthesis for
scan operations.</ipxact:description>
                              <ipxact:wire>
                                      <ipxact:direction>in</ipxact:direction>

```
                                </ipxact:wire>
                        </ipxact:port>
                        <ipxact:port>
                                <ipxact:name>csmea_cse_fscan_latchopen</ipxact:name>
                                <ipxact:description>Fabric Scan Latch Open Enable: This
signal controls the latch open during scan operations. This signal is bused to support
hard-IP block modular physical layer that requires scan control per lane.

Use of a bit vector for this signal name is optional. The value of ScanCtlWidth is
implementation-dependent and may vary per-agent or per-lane for a hard-IP IO agent.

Note:If this IP-block contains latches then this signal must be used to control them.
</ipxact:description>
                                <ipxact:wire>
                                        <ipxact:direction>in</ipxact:direction>
                                </ipxact:wire>
                        </ipxact:port>
                        <ipxact:port>
                                <ipxact:name>csmea_cse_fscan_latchclosed_b</ipxact:name>
                                <ipxact:description>Fabric Scan Latch Closed Bar: This
signal controls the latch closed during scan operations. This signal is bused to support
hard-IP block modular physical layer that requires scan control per lane.

Use of a bit vector for this signal name is optional. The value of ScanCtlWidth is
implementation-dependent and may vary per-agent or per-lane for a hard-IP IO agent.

Note:If this IP-block contains latches then this signal must be used to control them.
</ipxact:description>
                                <ipxact:wire>
                                        <ipxact:direction>in</ipxact:direction>
                                </ipxact:wire>
                        </ipxact:port>
                        <ipxact:port>
                                <ipxact:name>csmea_cse_fscan_clkgenctrl</ipxact:name>
                                <ipxact:description>Fabric Scan Clock Generator Control:
This signal bus overrides clock control values within the agent. The override value is
enabled with “fscan_clkgenctrlen”. This bus may be composed of clock select override and
other miscellaneous control signals used for conditioning the clock selects. For agents
with a TAP, these signals would be connected to the output of an assigned test data
register. For agents without a TAP, this signal bus is connected to a scan control logic
block (UCC /URC ) within the DFx fabric.

Note:This signal may be a single bit.</ipxact:description>
                                <ipxact:wire>
                                        <ipxact:direction>in</ipxact:direction>
                                </ipxact:wire>
                        </ipxact:port>
                        <ipxact:port>
                                <ipxact:name>csmea_cse_fscan_clkgenctrlen</ipxact:name>
                                <ipxact:description>Fabric Scan Clock Generator Control
Enable: This signal (or signal group) is the enable for the fscan_clkgenctrl override
control bus. A mux override control can manipulate the signal only during scan
operations. For agents with a TAP, these signals would be connected to the output of an
assigned test data register. For agents without a TAP, this signal group is a bus that is
connected to and controlled by the ULTiScan Clock Control (UCC) logic block.

If this signal is a bus, then bit[0] of the interface is assigned to the UCC.

fscan_clkgenctrlen[0]: This bit may be assigned to select between internal functional
clocks and external UCC clocks.

It is implementation dependent to bus this signal or use it as a single bit.
</ipxact:description>
                                <ipxact:wire>
                                        <ipxact:direction>in</ipxact:direction>
                                </ipxact:wire>
                        </ipxact:port>
```

```
				<ipxact:port>
					<ipxact:name>csmea_cse_fscan_mode_postscc</ipxact:name>
					<ipxact:description>Fabric Scan Mode for Post SCC control:
This signal is used to control the post SCC clock control mux. This signal is used for
hard-IP applications that require it. It is not expected to be used for soft IPs.
</ipxact:description>
					<ipxact:wire>
						<ipxact:direction>in</ipxact:direction>
					</ipxact:wire>
				</ipxact:port>
				<ipxact:port>
					<ipxact:name>FSCAN_INTEST_NOT_SUPPORTED</ipxact:name>
					<ipxact:description>Fabric Scan (sealing) Intest: This
signal indicates that internal border sealing is enabled. This signal is used for hard-IP
applications that require it. It is not expected to be used for soft
IPs.</ipxact:description>
					<ipxact:wire>
						<ipxact:direction>in</ipxact:direction>
					</ipxact:wire>
				</ipxact:port>
				<ipxact:port>
					<ipxact:name>FSCAN_EXTEST_NOT_SUPPORTED</ipxact:name>
					<ipxact:description>Fabric Scan (sealing) Extest: This
signal indicates that external border sealing is enabled. This signal is used for hard-IP
applications that require it. It is not expected to be used for soft
IPs.</ipxact:description>
					<ipxact:wire>
						<ipxact:direction>in</ipxact:direction>
					</ipxact:wire>
				</ipxact:port>
				<ipxact:port>
					<ipxact:name>fscan_ret_ctrl</ipxact:name>
					<ipxact:description>Fabric Scan Retention Control: This
signal determines the state of the retention cell within a retention flop for scan
operations. A mux in the Power Gate Common Block (PGCB) is controlled by fscan_mode. When
enabled, the signal is controlled from an ULTiScan Asynchronous Scan Controller (UASC) in
the DFx fabric.

Note:This signal is required if the IP-block supports retention cells.
</ipxact:description>
					<ipxact:wire>
						<ipxact:direction>in</ipxact:direction>
					</ipxact:wire>
				</ipxact:port>
				<ipxact:port>
					<ipxact:name>csmea_cse_fscan_isol_ctrl</ipxact:name>
					<ipxact:description>Fabric Scan Isolation Control: This
signal override the isolation control signal for improved scan coverage for the isolation
logic. This signal is required if this IP (SIP or HIP) supports isolation
logic.</ipxact:description>
					<ipxact:wire>
						<ipxact:direction>in</ipxact:direction>
					</ipxact:wire>
				</ipxact:port>
				<ipxact:port>
					<ipxact:name>csmea_cse_fscan_isol_lat_ctrl</ipxact:name>
					<ipxact:description>Fabric Scan Isolation Control: This
signal override the isolation control signal for improved scan coverage for the isolation
logic. This signal is required if this IP (SIP or HIP) supports latch based isolation
logic. </ipxact:description>
					<ipxact:wire>
						<ipxact:direction>in</ipxact:direction>
					</ipxact:wire>
				</ipxact:port>
				<ipxact:port>
					<ipxact:name>csmea_cse_fscan_shiften</ipxact:name>
```

```
                              <ipxact:description>Fabric Scan Shift Enable: This signal
determines whether the data chains are enabled for shifting. This does not apply to the
control chains. This signal is bused to support hard IP-block modular physical layer that
requires scan control per lane.

Use of a bit vector for this signal name is optional. The value of ScanCtlWidth is
implementation-dependent and may vary per-agent or per-lane for a hard-IP IO
agent.</ipxact:description>
                              <ipxact:wire>
                                     <ipxact:direction>in</ipxact:direction>
                              </ipxact:wire>
                       </ipxact:port>
                       <ipxact:port>
                              <ipxact:name>FSCAN_SLOS_EN_NOT_SUPPORTED</ipxact:name>
                              <ipxact:description>Fabric Scan Launch on Shift Enable:
This signal improves coverage using a launch-on-shift technique instead of launch-on-
capture, which is generally used for classic stuck-at testing. This signal is required on
hard-IP interfaces and not expected for soft IPs. </ipxact:description>
                              <ipxact:wire>
                                     <ipxact:direction>in</ipxact:direction>
                              </ipxact:wire>
                       </ipxact:port>
                       <ipxact:port>

       <ipxact:name>FSCAN_TPI_CONTROL_EN_NOT_SUPPORTED</ipxact:name>
                              <ipxact:description>Fabric Scan Test Point Insertion
Control Enable: This signal enables the inserted test point control for scan operations.
This signal is required on hard-IP interfaces and not expected for soft IPs.
</ipxact:description>
                              <ipxact:wire>
                                     <ipxact:direction>in</ipxact:direction>
                              </ipxact:wire>
                       </ipxact:port>
                       <ipxact:port>

       <ipxact:name>FSCAN_TPI_OBSERVE_EN_NOT_SUPPORTED</ipxact:name>
                              <ipxact:description>Fabric Scan Test Point Insertion
Observe Enable: This signal enables the observation of the test point node for scan
operations. This signal is required on hard-IP interfaces and not expected for soft IPs.
</ipxact:description>
                              <ipxact:wire>
                                     <ipxact:direction>in</ipxact:direction>
                              </ipxact:wire>
                       </ipxact:port>
                       <ipxact:port>
                              <ipxact:name>csmea_cse_fscan_ram_bypsel</ipxact:name>
                              <ipxact:description>Fabric Scan RAM Bypass Select: This
signal selects the bypass path around the array to conduct scan operations on this type
of array test configuration.

Note: This signal is required for any IP-block's memory wrapper that contains an array,
RAM, FIFO, etc</ipxact:description>
                              <ipxact:wire>
                                     <ipxact:direction>in</ipxact:direction>
                              </ipxact:wire>
                       </ipxact:port>
                       <ipxact:port>
                              <ipxact:name>FSCAN_RAM_INIT_EN_NOT_SUPPORTED</ipxact:name>
                              <ipxact:description>Fabric Scan RAM Initialization Enable:
This signal controls the array initialization for scan operations.

0: Normal operation
1: Enable initialization and the logic value currently driven on fscan_ram_init_val is
active.</ipxact:description>
                              <ipxact:wire>
                                     <ipxact:direction>in</ipxact:direction>
                              </ipxact:wire>
```

```
                        </ipxact:port>
                        <ipxact:port>
                                <ipxact:name>FSCAN_RAM_INIT_VAL_NOT_SUPPORTED</ipxact:name>
                                <ipxact:description>Fabric Scan RAM Initialization Value:
This signal is the RAM initialization value to control the DFx muxes with in the array
DFX_Wrapper.

0: Mux points to functional array controls. Also, this signal has no effect on all other
initialization logic.
1: Mux points to DFT array controls (from BIST) </ipxact:description>
                                <ipxact:wire>
                                        <ipxact:direction>in</ipxact:direction>
                                </ipxact:wire>
                        </ipxact:port>
                        <ipxact:port>
                                <ipxact:name>csmea_cse_fscan_ram_rddis_b</ipxact:name>
                                <ipxact:description>Fabric Scan RAM Read Disable Bar: This
signal controls the read enable on the agent’s array during scan operations.

Note: This signal is required for any IP-block's memory wrapper that contains an array,
RAM, FIFO, etc</ipxact:description>
                                <ipxact:wire>
                                        <ipxact:direction>in</ipxact:direction>
                                </ipxact:wire>
                        </ipxact:port>
                        <ipxact:port>
                                <ipxact:name>csmea_cse_fscan_ram_wrdis_b</ipxact:name>
                                <ipxact:description>Fabric Scan RAM Write Disable Bar: This
signal controls the write enable on the agent’s array during scan operations.

Note: This signal is required for any IP-block's memory wrapper that contains an array,
RAM, FIFO, etc</ipxact:description>
                                <ipxact:wire>
                                        <ipxact:direction>in</ipxact:direction>
                                </ipxact:wire>
                        </ipxact:port>
                </ipxact:ports>
        </ipxact:model>
        <ipxact:description></ipxact:description>
        <ipxact:vendorExtensions>
                <intel:component_attributes>
                        <intel:description></intel:description>
                        <intel:ip_metadata_version>1.9.3</intel:ip_metadata_version>
                        <intel:clk_domains>side_clk</intel:clk_domains>
                        <intel:voltage_domains>my_powerdomains</intel:voltage_domains>
                </intel:component_attributes>
        </ipxact:vendorExtensions>
</ipxact:component>
```