# Nifi功能列表

## 前言

为了创建有效的数据流，用户必须了解哪些类型的处理器可供他们使用。NiFi包含许多不同的处理器。这些处理器提供了从多种不同系统提取数据，路由，转换，处理，拆分和聚合数据以及将数据分配到多个系统的功能。

每个NiFi版本中都有不同的可用处理器。因此，在本功能列表我们不会尝试列出每个可用的处理器，但我们将重点介绍一些最常用的处理器，并根据它们的功能对它们进行分类。若需查看全部处理器，请查看https://nifi.apache.org/docs.html

## 数据转换

•CompressContent：压缩或解压缩内容

•ConvertCharacterSet：转换用于编码内容的字符集，字符集设置为另一个

•EncryptContent：加密或解密内容

•ReplaceText：使用正则表达式修改文字内容

•TransformXml：将XSLT转换应用于XML内容

•JoltTransformJSON：应用JOLT规范来转换JSON内容

## 路由和调解

•ControlRate：调节数据流量的一部分流量的速率。

•DetectDuplicate：根据用户定义的标准监控重复的FlowFiles。经常与HashContent一起使用

•DistributeLoad：通过仅分配一部分数据来加载余额或样本数据每个用户定义的关系。

•MonitorActivity：在用户定义的时间段内发送通知，而没有任何数据通过流中的特定点传入。可选择在数据流恢复时发送通知。

•RouteOnAttribute：根据它所包含的属性路由FlowFile。

•ScanAttribute：扫描FlowFile上用户定义的一组属性，检查是否任何属性都与用户定义的字典中找到的术语相匹配。

•RouteOnContent：搜索FlowFile的内容以查看它是否与用户定义的任何内容匹配

正则表达式。如果是，则将FlowFile路由到已配置的关系。

•ScanContent：搜索FlowFile的内容，查找用户定义的术语。

字典和路线的基础上存在或不存在这些条款。字典可以由文本条目或二进制条目组成。

•ValidateXml：根据XML模式验证XML内容;根据用户定义的XML Schema根据FlowFile的内容是否有效来路由FlowFile。

## 数据库访问

•ConvertJSONToSQL：将JSON文档转换为SQL INSERT或UPDATE命令，然后可以将其传递给PutSQL处理器

•ExecuteSQL：执行用户定义的SQL SELECT命令，将结果写入Avro格式的FlowFile

•PutSQL：通过执行由FlowFile内容定义的SQL DDM语句来更新数据库

•SelectHiveQL：针对Apache Hive数据库执行用户定义的HiveQL SELECT命令，将结果以Avro或CSV格式写入FlowFile

•PutHiveQL：通过执行由FlowFile的内容定义的HiveQL DDM语句来更新Hive数据库

## 属性提取

•EvaluateJsonPath：用户提供JSONPath表达式（与使用的XPath类似用于XML解析/提取），然后针对JSON评估这些表达式要么替换FlowFile内容，要么将值提取到用户命名的内容中

属性。

•EvaluateXPath：用户提供XPath表达式，然后这些表达式根据XML内容进行评估以替换FlowFile内容或提取值转换为用户名属性。

•EvaluateXQuery：用户提供一个XQuery查询，然后对这个查询进行评估将XML内容替换为FlowFile内容或将值提取到用户命名的属性。

•ExtractText：用户提供一个或多个正则表达式，然后进行评估针对FlowFile的文本内容，然后提取的值添加为用户命名的属性。

•HashAttribute：针对用户定义的串联执行散列函数现有属性列表。

•HashContent：针对FlowFile的内容执行哈希函数并添加散列值作为属性。

•IdentifyMimeType：评估FlowFile的内容以确定哪种类型FlowFile封装的文件。该处理器能够检测许多不同的MIME类型，如图像，文字处理器文档，文本和压缩格式仅举几个。

•UpdateAttribute：将任意数量的用户定义的属性添加或更新到FlowFile。这对添加静态配置的值以及派生属性值很有用动态使用表达式语言。该处理器还提供了“高级”用户界面“，允许用户根据用户提供的条件更新属性规则。

## 系统交互

•ExecuteProcess：运行用户定义的操作系统命令。 进程的StdOut被重定向，使得写入StdOut的内容成为出站FlowFile的内容。 此处理器是一个源处理器 - 其输出预计会生成一个新的FlowFile，并且系统调用预计不会接收输入。 为了向流程提供输入，请使用ExecuteStreamCommand处理器。

•ExecuteStreamCommand：运行用户定义的操作系统命令。该FlowFile的内容可以选择流式传输到进程的StdIn。 写入StdOut的内容将成为出站FlowFile的内容。 此处理器不能用于源处理器 - 它必须被馈入传入的FlowFiles才能执行其工作。 要使用源处理器执行相同类型的功能，请参阅ExecuteProcess处理器。

## 数据摄入

•GetFile：将文件内容从本地磁盘（或网络连接磁盘）流式传输到NiFi，然后删除原始文件。该处理器预计将文件从一个位置移动到另一个位置，并且不用于复制数据。

•GetFTP：通过FTP将远程文件的内容下载到NiFi中，然后删除该文件原始文件。预计该处理器将数据从一个位置移动到另一个位置位置，不能用于复制数据。

•GetSFTP：通过SFTP将远程文件的内容下载到NiFi中，然后删除原始文件。此处理器预计将数据从一个位置移动到另一个位置，不会用于复制数据。

•GetJMSQueue：从JMS队列下载消息，并根据JMS消息的内容创建FlowFile。 JMS属性也可以作为属性进行复制。

•GetJMSTopic：从JMS主题下载消息，并根据JMS消息的内容创建FlowFile。 JMS属性也可以作为属性进行复制。此处理器支持持久和非持久订阅。

•GetHTTP：将远程基于HTTP或HTTPS的URL的内容下载到NiFi中。处理器将记住ETag和最后修改日期，以确保数据不被持续摄取。

•ListenHTTP：启动HTTP（或HTTPS）服务器并侦听传入的连接。对于任何传入的POST请求，请求的内容将被写为FlowFile，并返回200响应。

•ListenUDP：侦听传入的UDP数据包，并为每个数据包或每个数据包数据包（取决于配置）创建一个FlowFile，并将FlowFile发送到成功关系。GetHDFS：监视HDFS中用户指定的目录。每当新文件进入HDFS时，它被复制到NiFi并从HDFS中删除。此处理器预计将移动文件从一个位置移动到另一个位置，不能用于复制数据。如果在群集内运行，该处理器也只能在主节点上运行。为了从HDFS复制数据并保留它，或者从群集中的多个节点中流式传输数据，请参阅ListHDFS处理器。

•ListHDFS / FetchHDFS：ListHDFS监视HDFS中的用户指定目录，并为每个遇到的文件发出一个包含文件名的FlowFile。然后它通过分布式缓存在整个NiFi集群中保持这种状态。然后，这些FlowFiles可以跨群集扇出并发送到FetchHDFS处理器，该处理器负责获取这些文件的实际内容并发出包含从HDFS获取的内容的FlowFiles。

•FetchS3Object：从Amazon Web Services（AWS）简单存储服务（S3）获取对象的内容。出站FlowFile包含从S3接收的内容。

•GetKafka：从Apache Kafka获取消息，专门用于0.8.x版本。消息可以作为每个消息的FlowFile发送，也可以使用用户指定的分隔符一起批量发送。

•GetMongo：对MongoDB执行用户指定的查询并将内容写入新的FlowFile。

•GetTwitter：允许用户注册一个过滤器来收听Twitter的“花园软管”或企业端点，为每个收到的推文创建一个FlowFile。

## 数据出口/发送数据

•PutEmail：将电子邮件发送到配置的收件人。 FlowFile的内容可以作为附件发送。

•PutFile：将FlowFile的内容写入本地（或网络）的目录

附件）文件系统。

•PutFTP：将FlowFile的内容复制到远程FTP服务器。

•PutSFTP：将FlowFile的内容复制到远程SFTP服务器。

•PutJMS：将FlowFile的内容作为JMS消息发送给JMS代理，可以根据属性选择添加JMS Properties。

•PutSQL：将FlowFile的内容作为SQL DDL语句（INSERT，UPDATE或DELETE）执行。 FlowFile的内容必须是有效的SQL语句。属性可以用作参数，以便FlowFile的内容可以是参数化的SQL语句，以避免SQL注入攻击。

•PutKafka：将FlowFile的内容作为消息发送给Apache Kafka，专门用于0.8.x版本。 FlowFile可以作为单个消息或分隔符（例如可以指定换行符）发送，以便为单个FlowFile发送许多消息。

•PutMongo：将FlowFile的内容作为INSERT或UPDATE发送到Mongo。

•ConsumeKafka\_0\_10：使用专门针对Kafka 0.10.x Consumer API构建的Apache Kafka消息。用于发送消息的互补NiFi处理器是PublishKafka\_0\_10。

•ConsumeKafkaRecord\_0\_10：使用专门针对Kafka 0.10.x Consumer API构建的Apache Kafka消息。用于发送消息的互补NiFi处理器是PublishKafkaRecord\_0\_10。请注意，目前，处理器假定从给定分区检索到的所有记录具有相同的模式。如果有任何Kafka消息被拉出但无法用配置的记录读取器或记录写入器进行分析或写入，则消息的内容将被写入单独的FlowFile，并且该FlowFile将被转移到'parse.failure'关系。否则，每个FlowFile都会发送到'成功'关系，并且可能会在单个FlowFile中包含许多单独的消息。添加'record.count'属性以指示FlowFile中包含的记录数量。

•PublishKafka\_0\_10：使用Kafka 0.10.x Producer API将FlowFile的内容作为消息发送给Apache Kafka。要发送的消息可以是单独的FlowFiles，也可以使用用户指定的分隔符（如换行符）进行分隔。用于获取消息的互补NiFi处理器是ConsumeKafka\_0\_10。

•PublishKafkaRecord\_0\_10：使用Kafka 0.10.x Producer API将FlowFile的内容作为单独记录发送到Apache Kafka。预计FlowFile的内容是面向记录的数据，可以由配置的记录读取器读取。用于获取消息的互补NiFi处理器是ConsumeKafka\_0\_10\_Record。

## 分裂和聚合

•SplitText：SplitText接受一个FlowFile，其内容为文本，并根据配置的行数将其拆分为1个或多个FlowFiles。 例如，处理器可以配置为将FlowFile拆分为许多FlowFiles，每个FlowFile只有1行。

•SplitJson：允许用户将包含数组或多个子对象的JSON对象拆分为每个JSON元素的FlowFile。

•SplitXml：允许用户将XML消息拆分为许多FlowFile，每个FlowFile都包含一段原始文件。这通常在多个XML元素与“包装器”元素结合在一起时使用。该处理器然后允许将这些元素分解成单独的XML元素。

•UnpackContent：解压缩不同类型的存档格式，例如ZIP和TAR。然后，档案中的每个文件将作为单个FlowFile传输。

•MergeContent：该处理器负责将多个FlowFiles合并为一个FlowFile。 FlowFiles可以通过将它们的内容连同可选的页眉，页脚和分界符或指定归档格式（如ZIP或TAR）合并在一起。 FlowFiles可以基于一个通用属性进行分类，或者如果它们被其他分割进程分割，则可以进行“碎片整理”。每个容器的最小和最大尺寸都是用户指定的，基于FlowFiles内容的元素数量或总大小，并且还可以指定可选的Timeout，以便FlowFiles只会等待其容器变满多少时间。

•SegmentContent：根据某些配置的数据大小将FlowFile分割成可能很多较小的FlowFiles。分割不是针对任何类型的分界符而是基于字节偏移量来执行的。这在传输FlowFiles之前使用，以便通过并行发送多个不同的片段来提供更低的延迟。另一方面，这些FlowFiles可以由MergeContent处理器使用碎片整理模式重新组合。

•SplitContent：将单个FlowFile拆分为可能的多个FlowFiles，与之类似SegmentContent。但是，对于SplitContent，分割不是在任意字节边界上执行，而是指定一个字节序列用于分割内容。

## HTTP

•GetHTTP：将远程基于HTTP或HTTPS的URL的内容下载到NiFi中。处理器将记住ETag和最后修改日期，以确保数据不被持续摄取。

•ListenHTTP：启动HTTP（或HTTPS）服务器并侦听传入的连接。对于任何传入的POST请求，请求的内容将被写为FlowFile，并返回200响应。

•调用HTTP：执行由用户配置的HTTP请求。这个处理器比GetHTTP和PostHTTP更通用，但需要更多的配置。此处理器不能用作源处理器，并且需要传入FlowFiles才能被触发以执行其任务。

•PostHTTP：执行HTTP POST请求，将FlowFile的内容作为消息的主体发送。这通常与ListenHTTP结合使用，以便在不能使用站点到站点的情况下（例如，当节点不能直接访问彼此并且能够通过HTTP进行通信时）在两个不同的NiFi实例之间传输数据代理）。注意：除了现有的RAW套接字传输外，HTTP还可用作站点到站点传输协议。它也支持HTTP代理。建议使用HTTP站点到站点，因为它具有更高的可伸缩性，并且可以使用输入/输出端口提供更好的用户身份验证和授权的双向数据传输。

•HandleHttpRequest / HandleHttpResponse：HandleHttpRequest处理器是与ListenHTTP类似，启动嵌入式HTTP（S）服务器的源处理器。但是，它不会向客户端发送响应。 相反，FlowFile与HTTP请求的主体一起作为其内容和属性发送给所有典型的Servlet参数，标题等作为属性。 HandleHttpResponse然后能够在FlowFile完成处理后将响应发送回客户端。 这些处理器总是需要相互结合使用，并允许用户在NiFi中可视地创建Web服务。 这对于向基于非基于网络的协议添加前端特别有用，或者在NiFi已经执行的某些功能（如数据格式转换）的基础上添加简单的Web服务。

## Amazon Web Services

•FetchS3Object：获取存储在Amazon Simple Storage Service（S3）中的对象的内容。 然后将从S3中检索的内容写入FlowFile的内容。

•PutS3Object：使用。将FlowFile的内容写入Amazon S3对象配置的凭据，密钥和存储桶名称。

•PutSNS：将FlowFile的内容作为通知发送到Amazon Simple Notification Service（SNS）。

•GetSQS：从Amazon Simple Queuing Service（SQS）中提取消息，并将消息的内容写入FlowFile的内容。

•PutSQS：将FlowFile的内容作为消息发送到Amazon Simple Queuing Service（SQS）。

•DeleteSQS：从Amazon Simple Queuing Service（SQS）中删除消息。 这可以与GetSQS一起使用，以便接收来自SQS的消息，对其执行一些处理，并且只有在成功完成处理后才从队列中删除该对象。